

(11)特許出願公開番号  
特開2002-278838  
(P2002-278838A)

(43)公開日 平成14年9月27日(2002.9.27)

(51)Int.Cl. <sup>7</sup>	識別記号	F I	ページ* (参考)
G 0 6 F 12/14	3 1 0	G 0 6 F 12/14	3 1 0 K 5 B 0 1 7
	3 2 0		3 2 0 A 5 B 0 3 5
12/00	5 3 7	12/00	5 3 7 A 5 B 0 5 8
15/00	3 3 0	15/00	3 3 0 A 5 B 0 8 2
G 0 6 K 17/00		G 0 6 K 17/00	S 5 B 0 8 5
審査請求 未請求 請求項の数46 O L (全 165 頁) 最終頁に続く			

(21)出願番号 特願2001-73352(P2001-73352)

(22)出願日 平成13年3月15日(2001.3.15)

(71)出願人 000002185

ソニー株式会社

東京都品川区北品川6丁目7番35号

(72)発明者 吉野 賢治

東京都品川区北品川6丁目7番35号 ソニ  
一株式会社内

(72) 発明者 石橋 義人

東京都品川区北品川6丁目7番35号 ソニ  
一株式会社内

(74) 代理人 100101801

弁理士 山田 英治 (外2名)

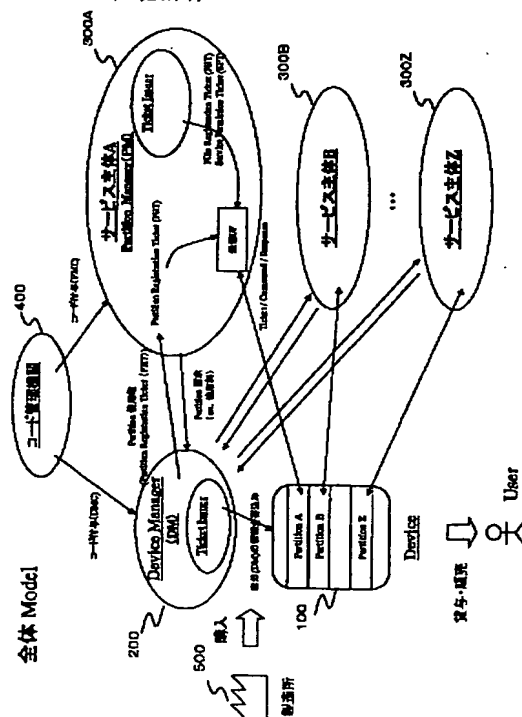
**最終頁に続く**

(54)【発明の名称】 メモリアクセス制御システム、デバイス管理装置、パーティション管理装置、メモリ搭載デバイス、およびメモリアクセス制御方法、並びにプログラム記憶媒体

(57) 【要約】

【課題】 デバイス内に生成した分割メモリ領域であるパーティションの独立した管理構成を可能としたメモリアクセス制御システムを提供する。

【解決手段】 複数のパーティションに分割されたメモリ領域のアクセスに対して、様々な種類のアクセス制御チケットを各デバイスまたはパーティションマネージャの管理の下に発行し、各チケットに記述されたルールに基づく処理をメモリ搭載デバイスにおいて実行する。メモリ部はパーティションマネージャによって管理されるメモリ領域としてのパーティション領域、デバイスマネージャによって管理されるデバイスマネージャ管理領域とを有し、パーティション対応の認証、デバイス対象の認証を公開鍵、共通鍵のいずれかに指定方式に従って実行することが可能とした。



**【特許請求の範囲】**

【請求項1】データファイルを格納したメモリ部を有するメモリ搭載デバイスに対するメモリアクセス制御システムであり前記メモリ搭載デバイスのメモリ部は、前記データファイルを格納し、パーティションマネージャによって管理されるメモリ領域としての1以上のパーティション領域と、該メモリ搭載デバイスの管理者としてのデバイスマネージャによって管理されるデバイスマネージャ管理領域とを有し、前記メモリ搭載デバイスは、前記メモリ部に対するアクセス制御チケットとして、前記デバイスマネージャの管理するアクセス制御チケット、または前記パーティションマネージャの管理するアクセス制御チケットをアクセス機器から受領し、受領チケットの記述に応じて処理を実行する構成を有することを特徴とするメモリアクセス制御システム。

【請求項2】前記アクセス制御チケットは、前記メモリ搭載デバイスとチケットを出力したアクセス機器間において実行すべき相互認証態様を指定した相互認証指定データを含み、前記メモリ搭載デバイスは、該アクセス制御チケットの相互認証指定データに応じた相互認証を実行し、認証の成立を条件として受領チケットの記録に応じた処理を実行する構成を有することを特徴とする請求項1に記載のメモリアクセス制御システム。

【請求項3】前記アクセス制御チケットは、前記メモリ搭載デバイスの受領したアクセス制御チケットの検証態様を指定したチケット検証指定データを含み、前記メモリ搭載デバイスは、該アクセス制御チケットのチケット検証指定データに応じたチケット検証処理を実行し、検証の成立を条件として受領チケットの記録に応じた処理を実行する構成を有することを特徴とする請求項1に記載のメモリアクセス制御システム。

【請求項4】前記アクセス制御チケットは、該アクセス制御チケットの発行手段のカテゴリまたは識別子を含み、前記メモリ搭載デバイスは、アクセス機器から受領したアクセス制御チケットに記述された該アクセス制御チケットの発行手段のカテゴリまたは識別子に基づいて、チケットが正当な発行手段により発行されたチケットであることの確認処理を実行し、該確認を条件として受領チケットの記録に応じた処理を実行する構成を有することを特徴とする請求項1に記載のメモリアクセス制御システム。

【請求項5】前記アクセス制御チケットは、該アクセス制御チケットの利用手段であるアクセス機器のカテゴリまたは識別子を含み、

前記メモリ搭載デバイスは、

アクセス機器から受領したアクセス制御チケットに記述された該アクセス制御チケットの利用手段であるアクセス機器のカテゴリまたは識別子に基づいて、チケットが正当な利用手段により提供されたチケットであることの確認処理を実行し、該確認を条件として受領チケットの記録に応じた処理を実行する構成を有することを特徴とする請求項1に記載のメモリアクセス制御システム。

【請求項6】前記デバイスマネージャの管理するアクセス制御チケットには、前記メモリ搭載デバイスのメモリ部に対するパーティションの生成処理または削除処理を許容するパーティション登録チケット（PRT）を含み、

前記メモリ搭載デバイスは、前記アクセス機器からパーティション登録チケット（PRT）を受領した場合は、受領パーティション登録チケット（PRT）の記録に従ったパーティションの生成処理または削除処理を実行することを特徴とする請求項1に記載のメモリアクセス制御システム。

【請求項7】前記パーティション登録チケット（PRT）は、前記デバイスマネージャの管理するチケット発行手段から前記パーティションマネージャの管理するチケットユーザとしてのアクセス機器に対して発行される構成であることを特徴とする請求項6に記載のメモリアクセス制御システム。

【請求項8】前記パーティションマネージャの管理するアクセス制御チケットには、前記メモリ搭載デバイスのメモリ部内に生成されたパーティション内に対するデータファイルの生成処理または削除処理を許容するファイル登録チケット（FRT）を含み、

前記メモリ搭載デバイスは、前記アクセス機器からファイル登録チケット（FRT）を受領した場合は、受領ファイル登録チケット（FRT）の記録に従ったファイルの生成処理または削除処理を実行することを特徴とする請求項1に記載のメモリアクセス制御システム。

【請求項9】前記ファイル登録チケット（FRT）は、前記パーティションマネージャの管理するチケット発行手段から前記パーティションマネージャの管理するチケットユーザとしてのアクセス機器に対して発行される構成であることを特徴とする請求項8に記載のメモリアクセス制御システム。

【請求項10】前記パーティションマネージャの管理するアクセス制御チケットには、前記メモリ搭載デバイスのメモリ部内のパーティション内のデータファイルに対するアクセスを許容するサービス許可チケット（SPT）を含み、前記メモリ搭載デバイスは、

前記アクセス機器からサービス許可チケット（SPT）を受領した場合は、  
受領サービス許可チケット（SPT）の記録に従ったデータファイルに対するアクセス処理を実行することを特徴とする請求項1に記載のメモリアクセス制御システム。

【請求項11】前記サービス許可チケット（SPT）は、前記パーティションマネージャの管理するチケット発行手段から前記パーティションマネージャの管理するチケットユーザとしてのアクセス機器に対して発行される構成であることを特徴とする請求項10に記載のメモリアクセス制御システム。

【請求項12】前記デバイスマネージャまたは前記パーティションマネージャの管理するアクセス制御チケットには、

前記メモリ搭載デバイスのメモリ部内の格納データの更新処理を許容するデータアップデートチケット（DUT）を含み、

前記メモリ搭載デバイスは、

前記アクセス機器からデータアップデートチケット（DUT）を受領した場合は、

受領データアップデートチケット（DUT）の記録に従ったデータ更新処理を実行することを特徴とする請求項1に記載のメモリアクセス制御システム。

【請求項13】前記デバイスマネージャの管理するデバイスマネージャ管理領域のデータ更新用のデータアップデートチケット（DUT）は、前記デバイスマネージャの管理するチケット発行手段から前記デバイスマネージャの管理するチケットユーザとしてのアクセス機器に対して発行され、

前記パーティションマネージャの管理するパーティション領域のデータ更新用のデータアップデートチケット（DUT）は、前記パーティションマネージャの管理するチケット発行手段から前記パーティションマネージャの管理するチケットユーザとしてのアクセス機器に対して発行される構成であることを特徴とする請求項12に記載のメモリアクセス制御システム。

【請求項14】データファイルを格納し、パーティション管理装置によって管理されるメモリ領域としての1以上のパーティション領域と、デバイス管理装置によって管理されるデバイスマネージャ管理領域とを有するメモリ搭載デバイスのデバイス管理を実行するデバイス管理装置であり、

前記メモリ搭載デバイスのメモリ部に対するパーティションの生成処理または削除処理を許容するメモリアクセス制御チケットとしてのパーティション登録チケット

（PRT）発行手段を有することを特徴とするデバイス管理装置。

【請求項15】前記デバイス管理装置は、

前記メモリ搭載デバイスに対する公開鍵証明書の発行管

理を実行する登録局構成を有することを特徴とする請求項14に記載のデバイス管理装置。

【請求項16】前記パーティション登録チケット（PRT）は、

前記メモリ搭載デバイスとチケットを出力したアクセス機器間において実行すべき相互認証態様を指定した相互認証指定データを含むことを特徴とする請求項14に記載のデバイス管理装置。

【請求項17】前記パーティション登録チケット（PRT）は、

前記メモリ搭載デバイスの受領したアクセス制御チケットの検証態様を指定したチケット検証指定データを含むことを特徴とする請求項14に記載のデバイス管理装置。

【請求項18】前記パーティション登録チケット（PRT）は、

該アクセス制御チケットの発行手段のカテゴリまたは識別子を含むことを特徴とする請求項14に記載のデバイス管理装置。

【請求項19】前記パーティション登録チケット（PRT）は、

該アクセス制御チケットの利用手段であるアクセス機器のカテゴリまたは識別子を含むことを特徴とする請求項14に記載のデバイス管理装置。

【請求項20】データファイルを格納し、パーティション管理装置によって管理されるメモリ領域としての1以上のパーティション領域と、デバイス管理装置によって管理されるデバイスマネージャ管理領域とを有するメモリ搭載デバイスのパーティション管理を実行するパーティション管理装置であり、

前記メモリ搭載デバイスのメモリ部に対して生成されたパーティション内に対するアクセスを許容するアクセス制御チケット発行手段を有することを特徴とするパーティション管理装置。

【請求項21】前記アクセス制御チケットは、前記メモリ搭載デバイスのメモリ部内に生成されたパーティション内に対するデータファイルの生成処理または削除処理を許容するファイル登録チケット（FRT）であることを特徴とする請求項20に記載のパーティション管理装置。

【請求項22】前記アクセス制御チケットは、前記メモリ搭載デバイスのメモリ部内のパーティション内のデータファイルに対するアクセスを許容するサービス許可チケット（SPT）であることを特徴とする請求項20に記載のパーティション管理装置。

【請求項23】前記パーティション管理装置は、前記メモリ搭載デバイスに対する公開鍵証明書の発行管理を実行する登録局構成を有することを特徴とする請求項20に記載のパーティション管理装置。

【請求項24】前記アクセス制御チケットは、

前記メモリ搭載デバイスとチケットを出力したアクセス機器間において実行すべき相互認証態様を指定した相互認証指定データを含むことを特徴とする請求項20に記載のパーティション管理装置。

【請求項25】前記アクセス制御チケットは、前記メモリ搭載デバイスの受領したアクセス制御チケットの検証態様を指定したチケット検証指定データを含むことを特徴とする請求項20に記載のパーティション管理装置。

【請求項26】前記アクセス制御チケットは、該アクセス制御チケットの発行手段のカテゴリまたは識別子を含むことを特徴とする請求項20に記載のパーティション管理装置。

【請求項27】前記アクセス制御チケットは、該アクセス制御チケットの利用手段であるアクセス機器のカテゴリまたは識別子を含むことを特徴とする請求項20に記載のパーティション管理装置。

【請求項28】データ格納可能なメモリ部を有するメモリ搭載デバイスであり、前記メモリ搭載デバイスのメモリ部は、パーティションマネージャによって管理されるメモリ領域としての1以上のパーティション領域と、該メモリ搭載デバイスの管理者としてのデバイスマネージャによって管理されるデバイスマネージャ管理領域とを有し、前記メモリ搭載デバイスは、前記メモリ部に対するアクセス制御チケットとして、前記デバイスマネージャの管理するアクセス制御チケット、または前記パーティションマネージャの管理するアクセス制御チケットをアクセス機器から受領し、受領チケットの記述に応じて処理を実行する制御手段を有することを特徴とするメモリ搭載デバイス。

【請求項29】前記アクセス制御チケットは、前記メモリ搭載デバイスとチケットを出力したアクセス機器間において実行すべき相互認証態様を指定した相互認証指定データを含み、前記制御手段は、該アクセス制御チケットの相互認証指定データに応じた相互認証を実行し、認証の成立を条件として受領チケットの記録に応じた処理を実行する構成を有することを特徴とする請求項28に記載のメモリ搭載デバイス。

【請求項30】前記アクセス制御チケットは、前記メモリ搭載デバイスの受領したアクセス制御チケットの検証態様を指定したチケット検証指定データを含み、前記制御手段は、該アクセス制御チケットのチケット検証指定データに応じたチケット検証処理を実行し、検証の成立を条件として受領チケットの記録に応じた処理を実行する構成を有することを特徴とする請求項28に記載のメモリ搭載デバイス。

【請求項31】前記アクセス制御チケットは、該アクセス制御チケットの発行手段のカテゴリまたは識別子を含み、

前記制御手段は、アクセス機器から受領したアクセス制御チケットに記述された該アクセス制御チケットの発行手段のカテゴリまたは識別子に基づいて、チケットが正当な発行手段により発行されたチケットであることの確認処理を実行し、該確認を条件として受領チケットの記録に応じた処理を実行する構成を有することを特徴とする請求項28に記載のメモリ搭載デバイス。

【請求項32】前記アクセス制御チケットは、該アクセス制御チケットの利用手段であるアクセス機器のカテゴリまたは識別子を含み、

前記制御手段は、アクセス機器から受領したアクセス制御チケットに記述された該アクセス制御チケットの利用手段であるアクセス機器のカテゴリまたは識別子に基づいて、チケットが正当な利用手段により提供されたチケットであることの確認処理を実行し、該確認を条件として受領チケットの記録に応じた処理を実行する構成を有することを特徴とする請求項28に記載のメモリ搭載デバイス。

【請求項33】データファイルを格納したメモリ部を有するメモリ搭載デバイスに対するメモリアクセス制御方法であり前記メモリ搭載デバイスのメモリ部は、前記データファイルを格納し、パーティションマネージャによって管理されるメモリ領域としての1以上のパーティション領域と、該メモリ搭載デバイスの管理者としてのデバイスマネージャによって管理されるデバイスマネージャ管理領域とを有し、前記メモリ搭載デバイスは、前記メモリ部に対するアクセス制御チケットとして、前記デバイスマネージャの管理するアクセス制御チケット、または前記パーティションマネージャの管理するアクセス制御チケットをアクセス機器から受領し、受領チケットの記述に応じて処理を実行することを特徴とするメモリアクセス制御方法。

【請求項34】前記アクセス制御チケットは、前記メモリ搭載デバイスとチケットを出力したアクセス機器間において実行すべき相互認証態様を指定した相互認証指定データを含み、

前記メモリ搭載デバイスは、該アクセス制御チケットの相互認証指定データに応じた相互認証を実行し、認証の成立を条件として受領チケットの記録に応じた処理を実行することを特徴とする請求項33に記載のメモリアクセス制御方法。

【請求項35】前記アクセス制御チケットは、前記メモリ搭載デバイスの受領したアクセス制御チケットの検証態様を指定したチケット検証指定データを含み、



前記メモリ搭載デバイスは、

該アクセス制御チケットのチケット検証指定データに応じたチケット検証処理を実行し、検証の成立を条件として受領チケットの記録に応じた処理を実行することを特徴とする請求項33に記載のメモリアクセス制御方法。

【請求項36】前記アクセス制御チケットは、該アクセス制御チケットの発行手段のカテゴリまたは識別子を含み、

前記メモリ搭載デバイスは、

アクセス機器から受領したアクセス制御チケットに記述された該アクセス制御チケットの発行手段のカテゴリまたは識別子に基づいて、チケットが正当な発行手段により発行されたチケットであることの確認処理を実行し、該確認を条件として受領チケットの記録に応じた処理を実行することを特徴とする請求項33に記載のメモリアクセス制御方法。

【請求項37】前記アクセス制御チケットは、

該アクセス制御チケットの利用手段であるアクセス機器のカテゴリまたは識別子を含み、

前記メモリ搭載デバイスは、

アクセス機器から受領したアクセス制御チケットに記述された該アクセス制御チケットの利用手段であるアクセス機器のカテゴリまたは識別子に基づいて、チケットが正当な利用手段により提供されたチケットであることの確認処理を実行し、該確認を条件として受領チケットの記録に応じた処理を実行することを特徴とする請求項33に記載のメモリアクセス制御方法。

【請求項38】前記デバイスマネージャの管理するアクセス制御チケットには、

前記メモリ搭載デバイスのメモリ部に対するパーティションの生成処理または削除処理を許容するパーティション登録チケット（PRT）を含み、

前記メモリ搭載デバイスは、

前記アクセス機器からパーティション登録チケット（PRT）を受領した場合は、

受領パーティション登録チケット（PRT）の記録に従ったパーティションの生成処理または削除処理を実行することを特徴とする請求項33に記載のメモリアクセス制御方法。

【請求項39】前記パーティション登録チケット（PRT）は、前記デバイスマネージャの管理するチケット発行手段から前記パーティションマネージャの管理するチケットユーザとしてのアクセス機器に対して発行されることを特徴とする請求項38に記載のメモリアクセス制御方法。

【請求項40】前記パーティションマネージャの管理するアクセス制御チケットには、前記メモリ搭載デバイスのメモリ部内に生成されたパーティション内に対するデータファイルの生成処理または削除処理を許容するファイル登録チケット（FRT）を

含み、

前記メモリ搭載デバイスは、

前記アクセス機器からファイル登録チケット（FRT）を受領した場合は、

受領ファイル登録チケット（FRT）の記録に従ったファイルの生成処理または削除処理を実行することを特徴とする請求項33に記載のメモリアクセス制御方法。

【請求項41】前記ファイル登録チケット（FRT）

は、前記パーティションマネージャの管理するチケット発行手段から前記パーティションマネージャの管理するチケットユーザとしてのアクセス機器に対して発行される構成であることを特徴とする請求項40に記載のメモリアクセス制御方法。

【請求項42】前記パーティションマネージャの管理するアクセス制御チケットには、

前記メモリ搭載デバイスのメモリ部内のパーティション内のデータファイルに対するアクセスを許容するサービス許可チケット（SPT）を含み、

前記メモリ搭載デバイスは、

前記アクセス機器からサービス許可チケット（SPT）を受領した場合は、

受領サービス許可チケット（SPT）の記録に従ったデータファイルに対するアクセス処理を実行することを特徴とする請求項33に記載のメモリアクセス制御方法。

【請求項43】前記サービス許可チケット（SPT）

は、前記パーティションマネージャの管理するチケット発行手段から前記パーティションマネージャの管理するチケットユーザとしてのアクセス機器に対して発行されることを特徴とする請求項42に記載のメモリアクセス制御方法。

【請求項44】前記デバイスマネージャまたは前記パーティションマネージャの管理するアクセス制御チケットには、

前記メモリ搭載デバイスのメモリ部内の格納データの更新処理を許容するデータアップデートチケット（DUT）を含み、

前記メモリ搭載デバイスは、

前記アクセス機器からデータアップデートチケット（DUT）を受領した場合は、

受領データアップデートチケット（DUT）の記録に従ったデータ更新処理を実行することを特徴とする請求項33に記載のメモリアクセス制御方法。

【請求項45】前記デバイスマネージャの管理するデバイスマネージャ管理領域のデータ更新用のデータアップデートチケット（DUT）は、前記デバイスマネージャの管理するチケット発行手段から前記デバイスマネージャの管理するチケットユーザとしてのアクセス機器に対して発行され、

前記パーティションマネージャの管理するパーティション領域のデータ更新用のデータアップデートチケット

(DUT)は、前記パーティションマネージャの管理するチケット発行手段から前記パーティションマネージャの管理するチケットユーザとしてのアクセス機器に対して発行されることを特徴とする請求項33に記載のメモリアccess制御方法。

【請求項46】データファイルを格納し、パーティションマネージャによって管理されるメモリ領域としての1以上のパーティション領域と、該メモリ搭載デバイスの管理者としてのデバイスマネージャによって管理されるデバイスマネージャ管理領域とを有するメモリ部を有するメモリ搭載デバイスに対するメモリアccess制御処理をコンピュータ・システム上で実行せしめるコンピュータ・プログラムを提供するプログラム記憶媒体であつて、前記コンピュータ・プログラムは、前記メモリ部に対するアクセス制御チケットとして、前記デバイスマネージャの管理するアクセス制御チケット、または前記パーティションマネージャの管理するアクセス制御チケットをアクセス機器から受領するステップと、アクセス機器との相互認証を実行するステップと、受領チケットの記述に応じたチケット検証処理を実行するステップと、受領チケットの記述に応じた処理を実行するステップと、を有することを特徴とするプログラム記憶媒体。

#### 【発明の詳細な説明】

##### 【0001】

【発明の属する技術分野】本発明は、メモリアccess制御システム、デバイス管理装置、パーティション管理装置、メモリ搭載デバイス、およびメモリアccess制御方法、並びにプログラム記憶媒体に関する。さらに、詳細には、1つのメモリを複数の領域（パーティション）に区分けし、各パーティション内にサービス提供者あるいは関係エンティティの管理するデータを格納して、ユーザが1つのメモリ搭載デバイスを用いて様々なサービスに供用することを可能としたメモリアccess制御システム、デバイス管理装置、パーティション管理装置、メモリ搭載デバイス、およびメモリアccess制御方法、並びにプログラム記憶媒体に関する。

##### 【0002】

【従来の技術】従来、メモリを保有するデバイスとしては、テープメディア、フロッピー（登録商標）ディスク、ハードディスク、光ディスク、半導体メディア等が利用されてきた。このうち、デバイス内のメモリをセキュアに管理できるものとして半導体メディアが注目されてきている。その理由は、半導体メモリは外部から容易にアクセスさせない構造、すなわち耐タンパ構造を実現しやすいからである。

【0003】耐タンパ構造は、例えばデバイスを半導体によるシングルチップ構成とし、該チップに制御部、メ

モリコントローラ、不揮発性メモリ、電圧検出手段、周波数検出手段等を備え、不揮発性メモリを外部から容易に読み書きができないようにアルミ層のようなダミー層に挟まれた構成とすることによって実現される。

【0004】このようなセキュアデバイスの従来のメモリ構造について図96「従来のメモリ構造」を用いて説明する。図96のメモリは、例えば電子マネーとして利用可能なメモリ構成を示している。図96に示すように、メモリ領域は大きく3つに別れている。すなわち、データ領域、メモリ管理領域、システム領域である。

【0005】データ領域には各データ内の先頭に格納された「データ構造」に基づくデータが格納されており、この例では、利用者名前、住所、電話番号、金額、メモ、ログの各データが格納される。メモリ管理領域には、データ領域の各データにアクセスするための格納アドレス、アクセス方法、アクセス認証鍵等が格納されている。例えば、データ領域のデータ1（利用者名前）のアクセスは読み出し（Read）のみが、アクセス認証鍵（0123……）の利用によって可能となることが示されている。また、システム領域には、デバイス識別子（ID）、データ領域にメモリ領域を確保するための認証鍵であるメモリ管理鍵等が格納される。

【0006】図96に示すメモリデバイスのデータ領域は複数に分割可能であり、これらの分割データ領域を、異なるサービス主体、例えば電子マネーであればそれぞれ別の電子マネーサービス提供主体（ex. 異なる銀行）が管理する構成とすることができる。各分割領域のデータは、個々のサービス提供主体の他、利用者、例えば電子マネーを利用した商品販売を行なう店舗に備えられたデバイスアクセス機器としてのリーダライタ（専用リーダライタまたはPCなど）によりデータの読み出し、書き込み、（ex. 残金データの更新）が実行される。

【0007】図96に示すような複数の分割されたデータ領域を持つセキュアデバイスの管理者と利用者の関係を図97「メモリ管理者・利用者」に示す。図97に示すように、セキュアデバイスの発行主体であるメモリ管理者と、このメモリ管理者からメモリ領域を割り当ててもらい、その割り当てられたメモリを利用するメモリ利用者がいる。メモリ利用者としてデータ1利用者～データ6利用者がいる。メモリ利用者とは例えば前述の電子マネーの例によれば、銀行または店舗等である。

【0008】メモリ管理者は、メモリ領域を確保するためのアクセスコントロール用のメモリ管理鍵を知っており、このメモリ管理鍵を利用して、それぞれのメモリ利用者のメモリ（分割データ領域）を割り当てる。また、メモリ利用者は各データ領域のデータにアクセスするためのアクセス認証鍵を知っており、このアクセス認証鍵を利用して、それぞれ割り当てられたデータ領域内のメモリにアクセスすることができる。アクセスの態様とし

てはデータの読み出し (Read)、書き込み (Write)、残金の減額 (Decrement) など、様々であり、それぞれの処理態様に応じてアクセス認証鍵を個別に設定して個別の処理の可否を設定することができる。

【0009】例えば図96に示すメモリ中のデータ4は、金額データであり、図97に示すようにデータ4の利用者はデータ4に対して減額 (Decrement) の処理と、読書き (Read/Write) の処理が可能である。図96の右下の表に示すように、データ4の減額 (Decrement) の処理と、読書き (Read/Write) の処理では、アクセスキーが異なり、各処理に対応したアクセスキーを使用してメモリにアクセスすることが必要となる。

【0010】図98に、メモリ管理者がメモリ利用者に対してメモリデバイス内のあるデータ領域を割り当てるメモリ確保処理を説明する図を示す。図98の「メモリの確保の方式」に示すように、メモリ管理者は、図の左側に示すメモリ確保用リーダ/ライタ (R/W: Reader/Writer) を用いて図の右側に示すメモリデバイスに対するデータ領域の確保処理を実行する。メモリ確保用リーダ/ライタ (R/W: Reader/Writer) には、メモリ管理鍵を保持するためのセキュアなNVRAM (Non-Volatile RAM) が備えられている。なお、メモリ確保用R/Wとしては、セキュアデバイスの専用の読み書きR/Wであっても、またセキュアデバイスがUSB、PCMCIAなどのI/Fを持つデバイスである場合、これらのインタフェースを介して読み書き可能な装置例えばPCであってもよい。

【0011】R/Wを用いてメモリを確保するためには、まずセキュアデバイスからデバイスIDを読み出す。次にR/W内において、メモリ管理鍵とデバイスIDを用いて認証鍵を生成し、生成した認証鍵を用いてセキュアデバイスと相互認証を実行する。相互認証処理は例えば共通鍵方式による相互認証 (ex. ISO/IEC9798-2) に従って実行される。

【0012】相互認証に成功した後、R/Wはデータ構造、データサイズ、アクセス方法、アクセス認証鍵をセッション鍵で暗号化し、必要に応じてデータ検証用のMAC (Message Authentication Code) 値を付加してセキュアデバイスにコマンドを送る。コマンドを受信したセキュアデバイスは、受信データを復号し、必要に応じてデータ改竄性の検証をMAC検証処理によって実行し、その後、受信データ内のデータサイズに応じてメモリのデータ領域にメモリ領域を確保し、確保した領域にデータ構造を書き込むとともに、メモリ管理領域に確保したメモリのアドレス、アクセス方法、アクセス認証鍵を書き込む。

【0013】このようにして、メモリデバイスには複数の分割データ領域が設定される。次に、図99の「メモ

リアクセス方法」に従って、複数の分割データ領域を持つメモリデバイスに対するメモリアクセス方法について説明する。図99の左側のリーダライタは、メモリ利用者の有するメモリアクセス用リーダライタ (R/W) であり、上述のメモリ確保用R/Wと同様、専用R/WあるいはPCなどで構成される。メモリアクセス用リーダライタ (R/W) には、アクセス認証鍵を保持するためのセキュアなNVRAMが備えられている。R/Wを用いてセキュアデバイスのデータ領域にアクセスするためには、まずセキュアデバイスからデバイスIDを読み出す。次にR/W内において、アクセス認証鍵とデバイスIDを用いて認証鍵を生成し、生成した認証鍵を用いてセキュアデバイスと相互認証を実行する。相互認証に成功した後、R/Wはアクセス認証鍵に対応するデータ領域のデータに所定のアクセスを行なう。

【0014】このときメモリ管理領域にはアクセス方法が規定されているため、例えば、図99の「メモリアクセス方法」に示すように、データ4 (金額データ) の減額 (Decrement) 用のアクセス認証に成功した場合は、データ4のデータの減額は可能であっても、加算、あるいは自由な書き換え処理はできない。このように認証処理に用いるアクセス認証鍵をそれぞれのアクセス態様に応じて異なる設定とすることにより各データの安全性を高めることができる。例えば減額処理用R/Wが盗難にあい、盗難にあった減額処理用R/W内のNVRAMが見破られた場合であっても、図99のセキュアデバイス内のデータ4 (金額データ) の不正な増加処理が行われる可能性を低減することができる。

【0015】一般に入金端末はATMと同様、セキュリティを高めることができるが、出金端末は店舗等で商品引き渡しの際の代金回収機として利用されることが多く、設置場所も様々であり、端末の盗難のリスクも高くセキュリティの度合いを高めることが困難である。従って、データアクセスに対してアクセス認証鍵を異ならせる構成が有効となる。

【0016】

【発明が解決しようとする課題】上述した従来の分割データ領域を持つメモリデバイスの利用形態において、メモリのデータ領域の確保処理、各データ領域のアクセス処理において、それぞれ、メモリ管理鍵を用いた認証処理、あるいはアクセス認証鍵を用いた認証処理を実行することによりそれぞれの処理を実行する構成としているが、これらは具体的には、例えばDES暗号アルゴリズムによる共通鍵を適用する構成であり、公開鍵方式による認証、あるいは公開鍵方式による検証を想定したものとはなっていない。

【0017】上述のようにメモリ管理鍵、アクセス認証鍵に共通鍵を適用した構成では認証およびアクセス許諾が一処理で実行されるという利点はあるが、認証鍵の漏洩により、漏洩鍵によるメモリアクセスが可能となって

しまうという欠点があり、セキュリティ上問題となる。

【0018】また、メモリデバイスに対するアクセスを実行するリーダライタ（R/W）の低コスト化を実現するために、リーダライタ（R/W）に暗号アルゴリズムを実装しない構成も想定されるが、このような構成とすれば、デバイス間との認証、通信データの暗号化の一切の処理が実行できず、ユーザの金額データ、その他ユーザのプライベート情報などを保持するデバイスに対するリーダライタとしては不適である。

【0019】本発明は、上述のような、従来技術の現状に鑑みてなされたものであり、複数のパーティションに分割されたメモリ領域のアクセスに対して、様々な種類のアクセス制御チケットを各デバイスまたはパーティション管理エンティティの管理の下に発行し、各チケットに記述されたルールに基づく処理をメモリ搭載デバイスにおいて実行する構成とすることにより、各パーティション内データの独立した管理構成を実現することを目的とする。

【0020】また、パーティション対応の認証、デバイス対象の認証を公開鍵、共通鍵のいずれか指定方式に従って実行することを可能とし、様々な環境下においてセキュアなデータ通信を実行可能としたメモリアクセス制御システム、デバイス管理装置、パーティション管理装置、メモリ搭載デバイス、およびメモリアクセス制御方法、並びにプログラム記憶媒体を提供することを目的とする。

【0021】

【課題を解決するための手段】本発明の第1の側面は、データファイルを格納したメモリ部を有するメモリ搭載デバイスに対するメモリアクセス制御システムであり前記メモリ搭載デバイスのメモリ部は、前記データファイルを格納し、パーティションマネージャによって管理されるメモリ領域としての1以上のパーティション領域と、該メモリ搭載デバイスの管理者としてのデバイスマネージャによって管理されるデバイスマネージャ管理領域とを有し、前記メモリ搭載デバイスは、前記メモリ部に対するアクセス制御チケットとして、前記デバイスマネージャの管理するアクセス制御チケット、または前記パーティションマネージャの管理するアクセス制御チケットをアクセス機器から受領し、受領チケットの記述に応じて処理を実行する構成を有することを特徴とするメモリアクセス制御システムにある。

【0022】さらに、本発明のメモリアクセス制御システムの一実施態様において、前記アクセス制御チケットは、前記メモリ搭載デバイスとチケットを出力したアクセス機器間において実行すべき相互認証態様を指定した相互認証指定データを含み、前記メモリ搭載デバイスは、該アクセス制御チケットの相互認証指定データに応じた相互認証を実行し、認証の成立を条件として受領チケットの記録に応じた処理を実行する構成を有すること

を特徴とする。

【0023】さらに、本発明のメモリアクセス制御システムの一実施態様において、前記アクセス制御チケットは、前記メモリ搭載デバイスの受領したアクセス制御チケットの検証態様を指定したチケット検証指定データを含み、前記メモリ搭載デバイスは、該アクセス制御チケットのチケット検証指定データに応じたチケット検証処理を実行し、検証の成立を条件として受領チケットの記録に応じた処理を実行する構成を有することを特徴とする。

【0024】さらに、本発明のメモリアクセス制御システムの一実施態様において、前記アクセス制御チケットは、該アクセス制御チケットの発行手段のカテゴリまたは識別子を含み、前記メモリ搭載デバイスは、アクセス機器から受領したアクセス制御チケットに記述された該アクセス制御チケットの発行手段のカテゴリまたは識別子に基づいて、チケットが正当な発行手段により発行されたチケットであることの確認処理を実行し、該確認を条件として受領チケットの記録に応じた処理を実行する構成を有することを特徴とする。

【0025】さらに、本発明のメモリアクセス制御システムの一実施態様において、前記アクセス制御チケットは、該アクセス制御チケットの利用手段であるアクセス機器のカテゴリまたは識別子を含み、前記メモリ搭載デバイスは、アクセス機器から受領したアクセス制御チケットに記述された該アクセス制御チケットの利用手段であるアクセス機器のカテゴリまたは識別子に基づいて、チケットが正当な利用手段により提供されたチケットであることの確認処理を実行し、該確認を条件として受領チケットの記録に応じた処理を実行する構成を有することを特徴とする。

【0026】さらに、本発明のメモリアクセス制御システムの一実施態様において、前記デバイスマネージャの管理するアクセス制御チケットには、前記メモリ搭載デバイスのメモリ部に対するパーティションの生成処理または削除処理を許容するパーティション登録チケット

（PRT）を含み、前記メモリ搭載デバイスは、前記アクセス機器からパーティション登録チケット（PRT）を受領した場合は、受領パーティション登録チケット

（PRT）の記録に従ったパーティションの生成処理または削除処理を実行することを特徴とする。

【0027】さらに、本発明のメモリアクセス制御システムの一実施態様において、前記パーティション登録チケット（PRT）は、前記デバイスマネージャの管理するチケット発行手段から前記パーティションマネージャの管理するチケットユーザとしてのアクセス機器に対して発行される構成であることを特徴とする。

【0028】さらに、本発明のメモリアクセス制御システムの一実施態様において、前記パーティションマネージャの管理するアクセス制御チケットには、前記メモリ

搭載デバイスのメモリ部内に生成されたパーティション内に対するデータファイルの生成処理または削除処理を許容するファイル登録チケット（FRT）を含み、前記メモリ搭載デバイスは、前記アクセス機器からファイル登録チケット（FRT）を受領した場合は、受領ファイル登録チケット（FRT）の記録に従ったファイルの生成処理または削除処理を実行することを特徴とする。

【0029】さらに、本発明のメモリアクセス制御システムの一実施態様において、前記ファイル登録チケット（FRT）は、前記パーティションマネージャの管理するチケット発行手段から前記パーティションマネージャの管理するチケットユーザとしてのアクセス機器に対して発行される構成であることを特徴とする。

【0030】さらに、本発明のメモリアクセス制御システムの一実施態様において、前記パーティションマネージャの管理するアクセス制御チケットには、前記メモリ搭載デバイスのメモリ部内のパーティション内のデータファイルに対するアクセスを許容するサービス許可チケット（SPT）を含み、前記メモリ搭載デバイスは、前記アクセス機器からサービス許可チケット（SPT）を受領した場合は、受領サービス許可チケット（SPT）の記録に従ったデータファイルに対するアクセス処理を実行することを特徴とする。

【0031】さらに、本発明のメモリアクセス制御システムの一実施態様において、前記サービス許可チケット（SPT）は、前記パーティションマネージャの管理するチケット発行手段から前記パーティションマネージャの管理するチケットユーザとしてのアクセス機器に対して発行される構成であることを特徴とする。

【0032】さらに、本発明のメモリアクセス制御システムの一実施態様において、前記デバイスマネージャまたは前記パーティションマネージャの管理するアクセス制御チケットには、前記メモリ搭載デバイスのメモリ部内の格納データの更新処理を許容するデータアップデートチケット（DUT）を含み、前記メモリ搭載デバイスは、前記アクセス機器からデータアップデートチケット（DUT）を受領した場合は、受領データアップデートチケット（DUT）の記録に従ったデータ更新処理を実行することを特徴とする。

【0033】さらに、本発明のメモリアクセス制御システムの一実施態様において、前記デバイスマネージャの管理するデバイスマネージャ管理領域のデータ更新用のデータアップデートチケット（DUT）は、前記デバイスマネージャの管理するチケット発行手段から前記デバイスマネージャの管理するチケットユーザとしてのアクセス機器に対して発行され、前記パーティションマネージャの管理するパーティション領域のデータ更新用のデータアップデートチケット（DUT）は、前記パーティションマネージャの管理するチケット発行手段から前記パーティションマネージャの管理するチケットユーザと

してのアクセス機器に対して発行される構成であることを特徴とする。

【0034】さらに、本発明の第2の側面は、データファイルを格納し、パーティション管理装置によって管理されるメモリ領域としての1以上のパーティション領域と、デバイス管理装置によって管理されるデバイスマネージャ管理領域とを有するメモリ搭載デバイスのデバイス管理を実行するデバイス管理装置であり、前記メモリ搭載デバイスのメモリ部に対するパーティションの生成処理または削除処理を許容するメモリアクセス制御チケットとしてのパーティション登録チケット（PRT）発行手段を有することを特徴とするデバイス管理装置にある。

【0035】さらに、本発明のデバイス管理装置の一実施態様において、前記メモリ搭載デバイスに対する公開鍵証明書の発行管理を実行する登録局構成を有することを特徴とする。

【0036】さらに、本発明のデバイス管理装置の一実施態様において、前記パーティション登録チケット（PRT）は、前記メモリ搭載デバイスとチケットを出力したアクセス機器間において実行すべき相互認証態様を指定した相互認証指定データを含むことを特徴とする。

【0037】さらに、本発明のデバイス管理装置の一実施態様において、前記パーティション登録チケット（PRT）は、前記メモリ搭載デバイスの受領したアクセス制御チケットの検証態様を指定したチケット検証指定データを含むことを特徴とする。

【0038】さらに、本発明のデバイス管理装置の一実施態様において、前記パーティション登録チケット（PRT）は、該アクセス制御チケットの発行手段のカテゴリまたは識別子を含むことを特徴とする。

【0039】さらに、本発明のデバイス管理装置の一実施態様において、前記パーティション登録チケット（PRT）は、該アクセス制御チケットの利用手段であるアクセス機器のカテゴリまたは識別子を含むことを特徴とする。

【0040】さらに、本発明の第3の側面は、データファイルを格納し、パーティション管理装置によって管理されるメモリ領域としての1以上のパーティション領域と、デバイス管理装置によって管理されるデバイスマネージャ管理領域とを有するメモリ搭載デバイスのパーティション管理を実行するパーティション管理装置であり、前記メモリ搭載デバイスのメモリ部に対して生成されたパーティション内に対するアクセスを許容するアクセス制御チケット発行手段を有することを特徴とするパーティション管理装置にある。

【0041】さらに、本発明のパーティション管理装置の一実施態様において、前記アクセス制御チケットは、前記メモリ搭載デバイスのメモリ部内に生成されたパーティション内に対するデータファイルの生成処理または

削除処理を許容するファイル登録チケット（FRT）であることを特徴とする。

【0042】さらに、本発明のパーティション管理装置の一実施態様において、前記アクセス制御チケットは、前記メモリ搭載デバイスのメモリ部内のパーティション内のデータファイルに対するアクセスを許容するサービス許可チケット（SPT）であることを特徴とする。

【0043】さらに、本発明のパーティション管理装置の一実施態様において、前記パーティション管理装置は、前記メモリ搭載デバイスに対する公開鍵証明書発行管理を実行する登録局構成を有することを特徴とする。

【0044】さらに、本発明のパーティション管理装置の一実施態様において、前記アクセス制御チケットは、前記メモリ搭載デバイスとチケットを出力したアクセス機器間において実行すべき相互認証態様を指定した相互認証指定データを含むことを特徴とする。

【0045】さらに、本発明のパーティション管理装置の一実施態様において、前記アクセス制御チケットは、前記メモリ搭載デバイスの受領したアクセス制御チケットの検証態様を指定したチケット検証指定データを含むことを特徴とする。

【0046】さらに、本発明のパーティション管理装置の一実施態様において、前記アクセス制御チケットは、該アクセス制御チケットの発行手段のカテゴリまたは識別子を含むことを特徴とする。

【0047】さらに、本発明のパーティション管理装置の一実施態様において、前記アクセス制御チケットは、該アクセス制御チケットの利用手段であるアクセス機器のカテゴリまたは識別子を含むことを特徴とする。

【0048】さらに、本発明の第4の側面は、データ格納可能なメモリ部を有するメモリ搭載デバイスであり、前記メモリ搭載デバイスのメモリ部は、パーティションマネージャによって管理されるメモリ領域としての1以上のパーティション領域と、該メモリ搭載デバイスの管理者としてのデバイスマネージャによって管理されるデバイスマネージャ管理領域とを有し、前記メモリ搭載デバイスは、前記メモリ部に対するアクセス制御チケットとして、前記デバイスマネージャの管理するアクセス制御チケット、または前記パーティションマネージャの管理するアクセス制御チケットをアクセス機器から受領し、受領チケットの記述に応じて処理を実行する制御手段を有することを特徴とするメモリ搭載デバイスにある。

【0049】さらに、本発明のメモリ搭載デバイスの一実施態様において、前記アクセス制御チケットは、前記メモリ搭載デバイスとチケットを出力したアクセス機器間において実行すべき相互認証態様を指定した相互認証指定データを含み、前記制御手段は、該アクセス制御チケットの相互認証指定データに応じた相互認証を実行

し、認証の成立を条件として受領チケットの記録に応じた処理を実行する構成を有することを特徴とする。

【0050】さらに、本発明のメモリ搭載デバイスの一実施態様において、前記アクセス制御チケットは、前記メモリ搭載デバイスの受領したアクセス制御チケットの検証態様を指定したチケット検証指定データを含み、前記制御手段は、該アクセス制御チケットのチケット検証指定データに応じたチケット検証処理を実行し、検証の成立を条件として受領チケットの記録に応じた処理を実行する構成を有することを特徴とする。

【0051】さらに、本発明のメモリ搭載デバイスの一実施態様において、前記アクセス制御チケットは、該アクセス制御チケットの発行手段のカテゴリまたは識別子を含み、前記制御手段は、アクセス機器から受領したアクセス制御チケットに記述された該アクセス制御チケットの発行手段のカテゴリまたは識別子に基づいて、チケットが正当な発行手段により発行されたチケットであることの確認処理を実行し、該確認を条件として受領チケットの記録に応じた処理を実行する構成を有することを特徴とする。

【0052】さらに、本発明のメモリ搭載デバイスの一実施態様において、前記アクセス制御チケットは、該アクセス制御チケットの利用手段であるアクセス機器のカテゴリまたは識別子を含み、前記制御手段は、アクセス機器から受領したアクセス制御チケットに記述された該アクセス制御チケットの利用手段であるアクセス機器のカテゴリまたは識別子に基づいて、チケットが正当な利用手段により提供されたチケットであることの確認処理を実行し、該確認を条件として受領チケットの記録に応じた処理を実行する構成を有することを特徴とする。

【0053】さらに、本発明の第5の側面は、データファイルを格納したメモリ部を有するメモリ搭載デバイスに対するメモリアクセス制御方法であり前記メモリ搭載デバイスのメモリ部は、前記データファイルを格納し、パーティションマネージャによって管理されるメモリ領域としての1以上のパーティション領域と、該メモリ搭載デバイスの管理者としてのデバイスマネージャによって管理されるデバイスマネージャ管理領域とを有し、前記メモリ搭載デバイスは、前記メモリ部に対するアクセス制御チケットとして、前記デバイスマネージャの管理するアクセス制御チケット、または前記パーティションマネージャの管理するアクセス制御チケットをアクセス機器から受領し、受領チケットの記述に応じて処理を実行することを特徴とするメモリアクセス制御方法にある。

【0054】さらに、本発明のメモリアクセス制御方法の一実施態様において、前記アクセス制御チケットは、前記メモリ搭載デバイスとチケットを出力したアクセス機器間において実行すべき相互認証態様を指定した相互認証指定データを含み、前記メモリ搭載デバイスは、該

アクセス制御チケットの相互認証指定データに応じた相互認証を実行し、認証の成立を条件として受領チケットの記録に応じた処理を実行することを特徴とする。

【0055】さらに、本発明のメモリアクセス制御方法の一実施態様において、前記アクセス制御チケットは、前記メモリ搭載デバイスの受領したアクセス制御チケットの検証態様を指定したチケット検証指定データを含み、前記メモリ搭載デバイスは、該アクセス制御チケットのチケット検証指定データに応じたチケット検証処理を実行し、検証の成立を条件として受領チケットの記録に応じた処理を実行することを特徴とする。

【0056】さらに、本発明のメモリアクセス制御方法の一実施態様において、前記アクセス制御チケットは、該アクセス制御チケットの発行手段のカテゴリまたは識別子を含み、前記メモリ搭載デバイスは、アクセス機器から受領したアクセス制御チケットに記述された該アクセス制御チケットの発行手段のカテゴリまたは識別子に基づいて、チケットが正当な発行手段により発行されたチケットであることの確認処理を実行し、該確認を条件として受領チケットの記録に応じた処理を実行することを特徴とする。

【0057】さらに、本発明のメモリアクセス制御方法の一実施態様において、前記アクセス制御チケットは、該アクセス制御チケットの利用手段であるアクセス機器のカテゴリまたは識別子を含み、前記メモリ搭載デバイスは、アクセス機器から受領したアクセス制御チケットに記述された該アクセス制御チケットの利用手段であるアクセス機器のカテゴリまたは識別子に基づいて、チケットが正当な利用手段により提供されたチケットであることの確認処理を実行し、該確認を条件として受領チケットの記録に応じた処理を実行することを特徴とする。

【0058】さらに、本発明のメモリアクセス制御方法の一実施態様において、前記デバイスマネージャの管理するアクセス制御チケットには、前記メモリ搭載デバイスのメモリ部に対するパーティションの生成処理または削除処理を許容するパーティション登録チケット（PRT）を含み、前記メモリ搭載デバイスは、前記アクセス機器からパーティション登録チケット（PRT）を受領した場合は、受領パーティション登録チケット（PRT）の記録に従ったパーティションの生成処理または削除処理を実行することを特徴とする。

【0059】さらに、本発明のメモリアクセス制御方法の一実施態様において、前記パーティション登録チケット（PRT）は、前記デバイスマネージャの管理するチケット発行手段から前記パーティションマネージャの管理するチケットユーザとしてのアクセス機器に対して発行されることを特徴とする。

【0060】さらに、本発明のメモリアクセス制御方法の一実施態様において、前記パーティションマネージャの管理するアクセス制御チケットには、前記メモリ搭載

デバイスのメモリ部内に生成されたパーティション内に対するデータファイルの生成処理または削除処理を許容するファイル登録チケット（FRT）を含み、前記メモリ搭載デバイスは、前記アクセス機器からファイル登録チケット（FRT）を受領した場合は、受領ファイル登録チケット（FRT）の記録に従ったファイルの生成処理または削除処理を実行することを特徴とする。

【0061】さらに、本発明のメモリアクセス制御方法の一実施態様において、前記ファイル登録チケット（FRT）は、前記パーティションマネージャの管理するチケット発行手段から前記パーティションマネージャの管理するチケットユーザとしてのアクセス機器に対して発行される構成であることを特徴とする。

【0062】さらに、本発明のメモリアクセス制御方法の一実施態様において、前記パーティションマネージャの管理するアクセス制御チケットには、前記メモリ搭載デバイスのメモリ部内のパーティション内のデータファイルに対するアクセスを許容するサービス許可チケット（SPT）を含み、前記メモリ搭載デバイスは、前記アクセス機器からサービス許可チケット（SPT）を受領した場合は、受領サービス許可チケット（SPT）の記録に従ったデータファイルに対するアクセス処理を実行することを特徴とする。

【0063】さらに、本発明のメモリアクセス制御方法の一実施態様において、前記サービス許可チケット（SPT）は、前記パーティションマネージャの管理するチケット発行手段から前記パーティションマネージャの管理するチケットユーザとしてのアクセス機器に対して発行されることを特徴とする。

【0064】さらに、本発明のメモリアクセス制御方法の一実施態様において、前記デバイスマネージャまたは前記パーティションマネージャの管理するアクセス制御チケットには、前記メモリ搭載デバイスのメモリ部内の格納データの更新処理を許容するデータアップデートチケット（DUT）を含み、前記メモリ搭載デバイスは、前記アクセス機器からデータアップデートチケット（DUT）を受領した場合は、受領データアップデートチケット（DUT）の記録に従ったデータ更新処理を実行することを特徴とする。

【0065】さらに、本発明のメモリアクセス制御方法の一実施態様において、前記デバイスマネージャの管理するデバイスマネージャ管理領域のデータ更新用のデータアップデートチケット（DUT）は、前記デバイスマネージャの管理するチケット発行手段から前記デバイスマネージャの管理するチケットユーザとしてのアクセス機器に対して発行され、前記パーティションマネージャの管理するパーティション領域のデータ更新用のデータアップデートチケット（DUT）は、前記パーティションマネージャの管理するチケット発行手段から前記パーティションマネージャの管理するチケットユーザとして

のアクセス機器に対して発行されることを特徴とする。

【0066】さらに、本発明の第6の側面は、データファイルを格納し、パーティションマネージャによって管理されるメモリ領域としての1以上のパーティション領域と、該メモリ搭載デバイスの管理者としてのデバイスマネージャによって管理されるデバイスマネージャ管理領域とを有するメモリ部を有するメモリ搭載デバイスに対するメモリアクセス制御処理をコンピュータ・システム上で実行せしめるコンピュータ・プログラムを提供するプログラム記憶媒体であって、前記コンピュータ・プログラムは、前記メモリ部に対するアクセス制御チケットとして、前記デバイスマネージャの管理するアクセス制御チケット、または前記パーティションマネージャの管理するアクセス制御チケットをアクセス機器から受領するステップと、アクセス機器との相互認証を実行するステップと、受領チケットの記述に応じたチケット検証処理を実行するステップと、受領チケットの記述に応じた処理を実行するステップと、を有することを特徴とするプログラム記憶媒体にある。

【0067】なお、本発明のプログラム記憶媒体は、例えば、様々なプログラム・コードを実行可能な汎用コンピュータ・システムに対して、コンピュータ・プログラムをコンピュータ可読な形式で提供する媒体である。媒体は、CDやFD、MOなどの記録媒体、通信可能媒体など、その形態は特に限定されない。

【0068】このようなプログラム記憶媒体は、コンピュータ・システム上で所定のコンピュータ・プログラムと記憶媒体との構造上又は機能上の協働的關係を定義したものである。換言すれば、該記憶媒体を介してコンピュータ・プログラムをコンピュータ・システムにインストールすることによって、コンピュータ・システム上では協働的作用が発揮され、本発明の他の側面と同様の作用効果を得ることができるのである。

【0069】本発明のさらに他の目的、特徴や利点は、後述する本発明の実施例や添付する図面に基づくより詳細な説明によって明らかになるであろう。なお、本明細書においてシステムとは、複数の装置の論理的集合構成であり、各構成の装置が同一筐体内にあるものには限らない。

【0070】

【発明の実施の形態】以下、図面を参照しながら、本発明の実施の形態について詳細に説明する。なお、説明は、以下の項目に従って行なう。

A. デバイスを利用したデータ処理システムの構成エンティティおよびチケットに関する説明

A1. メモリ搭載デバイスを利用したデータ管理システムの概要

A2. デバイスの構成

A3. デバイスマネージャの構成

A4. パーティションマネージャの構成

A5. チケットユーザ（デバイスアクセス機器としてのリーダーライター）の構成

A6. 公開鍵証明書

A7. デバイスのメモリ部における格納データ

A7. 1. デバイス固有情報およびデバイス内パーティション情報領域

A7. 2. パーティション領域

A8. 各チケットのデータフォーマット

A8. 1. パーティション登録チケット（PRT）

A8. 2. ファイル登録チケット（FRT）

A8. 3. サービス許可チケット（SPT）

A8. 4. データアップデートチケット（DUT）

B. ユーザに対するデバイスの配布、デバイスに対する各種設定、デバイス利用処理の詳細についての説明

B1. デバイス初期登録から利用までの流れ

B2. デバイス製造エンティティによる初期登録処理

B3. デバイスマネージャの管轄処理

B3. 1. デバイスマネージャによるデバイス登録処理

B3. 2. デバイスマネージャ管理下における公開鍵証明書発行処理

B4. パーティションマネージャの管轄処理

B4. 1. パーティションマネージャ管理下におけるパーティション登録チケット（PRT）を利用したパーティション設定登録、削除処理

B4. 2. パーティションマネージャ管理下における公開鍵証明書発行処理

B4. 3. パーティション生成処理各方式における処理手順

B4. 4. ファイル登録チケット（FRT）を利用したファイル生成、消去処理

B4. 5. ファイル生成処理各方式における処理手順

B4. 6. サービス許可チケット（SPT）を利用したサービス（ファイルアクセス）処理

B4. 7. サービス許可チケット（SPT）を利用したアクセス処理各方式における処理手順

B5. データアップデートチケット（DUT）を利用したデバイスのデータ更新処理

【0071】

【実施例】[A1. メモリ搭載デバイスを利用したデータ管理システムの概要] 図1に本発明のデータ管理システムの概要を説明する図を示す。メモリ搭載デバイス（以下デバイス）100はデバイス製造エンティティ（manufacturer）500により製造され、デバイス管理エンティティとしてのデバイスマネージャ（DM: Device Manager）200の管理の下にユーザに提供されて利用される。ユーザに対するデバイスの提供形態は、貸与または販売（譲渡を含む）など、いずれの形態でもよい。

【0072】デバイス100は、メモリ領域が複数のデ



ータ格納領域としてのパーティションに分割され、個々のパーティション(Partition A,B…Z)は、様々なサービス主体(A, B, …Z) 300A~300Zとしてのパーティションマネージャの管理の下、様々なサービスに利用される。

【0073】デバイス100に対するパーティションの設定登録処理、デバイスに設定されたパーティション内におけるファイルの設定登録処理、さらに、登録された各ファイルに対するアクセス処理にはそれぞれ正当なチケット発行手段(Ticket Issuer)の発行したデバイスに対するアクセスコントロールチケットを必要とする。

【0074】デバイス100に対するパーティションの設定登録処理には、正当なチケット発行手段(Ticket Issuer)の発行したパーティション登録チケット(PRT: Partition Registration Ticket)が必要であり、デバイスに設定されたパーティション内に対するファイルの設定登録処理には、正当なチケット発行手段(Ticket Issuer)の発行したファイル登録チケット(FRT: File Registration Ticket)が必要であり、また、各ファイルに対するアクセスには正当なチケット発行手段(Ticket Issuer)の発行したサービス許可チケット(SPT: Service Permission Ticket)が必要となる。

【0075】各チケットには、デバイス100に対するアクセスルール、例えばデバイスに対して読み書きなど各種処理を実行するリーダ/ライタとデバイス間の相互認証処理に関するルールの他、例えばパーティション登録チケット(PRT)であれば、設定できるパーティションサイズ、ファイル登録チケット(FRT)であれば、設定できるファイルサイズ、サービス許可チケット(SPT)であれば、実行可能なアクセス態様(ex. データ読み出し、書き込みなど)などが格納され、さらにチケット発行者、チケット利用者に関する情報、その他の情報が格納される。また、これらチケット格納データに対する改竄チェック用のICV(Integrity Check Value)が記録され、チケットの改竄の無いことを条件としてチケットに記録された範囲内の処理が実行可能となる。これらチケットの詳細については後段で説明する。

【0076】図1において示す例では、パーティション登録チケット(PRT)を発行するチケット発行手段(Ticket Issuer)はデバイスマネージャ(DM)200内に設定され、パーティションマネージャとしてのサービス主体A, 300A内にファイル登録チケット(FRT)、およびサービス許可チケット(SPT)を発行するチケット発行手段(Ticket Issuer)が設定される。なお図1の構成は、サービス主体B…Z, 300B~300Zについても基本的にサービス主体Aと同様の構成を持つものであり、各サービス主体にファイル登録チケット(FRT)、およびサービス許可チケット(S

PT)を発行するチケット発行手段(Ticket Issuer)が設定される。

【0077】なお、図1では、サービス主体とパーティションマネージャ(PM)を同一エンティティとして示しているが、必ずしもこれらのエンティティは同一であることは必要でなく、デバイスに設定されるメモリ領域としてのパーティションを管理するパーティションマネージャと、パーティションマネージャの管理するメモリ領域であるパーティションをパーティションマネージャから所定契約の下に借り受けて、借り受けたパーティション内に様々なファイルを格納してサービスを提供するサービス主体とが別エンティティとして存在してもよい。以下の説明では、説明を簡略化するためにサービス主体がパーティションマネージャとして機能する構成例について説明する。

【0078】各サービス主体300A~300Zとしてのパーティションマネージャ(PM)は、デバイスマネージャ(DM)200に対して、例えば相応の対価を支払うなど所定の契約の下に、パーティション登録チケット(PRT)の発行要求を行ない、デバイスマネージャ(DM)の許諾の下、デバイスマネージャ(DM)内のチケット発行手段(Ticket Issuer)が各サービス主体としてのパーティションマネージャ(PM)に対してパーティション登録チケット(PRT)を発行する。

【0079】各サービス主体(パーティションマネージャ(PM))300は、通信インタフェース(I/F)を介してユーザの所有デバイス100に対するアクセスを実行し、デバイスマネージャ(DM)200から受領したパーティション登録チケット(PRT)に記録されたルールに従った認証、検証等の処理を実行し、かつパーティション登録チケット(PRT)に記録された許可範囲内のパーティションの設定登録処理を実行する。この処理については後段で詳細に説明する。

【0080】通信I/Fは、有線、無線を問わず、外部機器(デバイス)とのデータ通信可能なインタフェースであればよく、例えば、デバイスがUSB接続構成を持つ場合はUSB I/F、また、ICカード型であればICカード用リーダライタ、さらに公衆回線、通信回線、インターネットなど各種の通信機能を持つデバイス、あるいはこれらの通信装置に接続可能なデバイスであれば、各通信方式に従ったデータ通信I/Fとして構成される。

【0081】また、デバイス100にサービス主体300のパーティションが設定されると、各サービス主体300は、通信インタフェース(I/F)を介してユーザ所有のデバイス100にアクセスし、各サービス主体300のチケット発行手段(Ticket Issuer)の発行するファイル登録チケット(FRT)に記録されたルールに従った認証、検証等の処理を実行し、かつファイル登録チケット(FRT)に記録された許可範囲内のファイル

の設定登録処理を実行する。この処理については後段で詳細に説明する。

【0082】さらに、各サービス主体300は、通信インタフェース(I/F)を介してユーザの所有デバイス100にアクセスし、各サービス主体のチケット発行手段(Ticket Issuer)の発行するサービス許可チケット(SPT)に記録されたルールに従った認証、検証等の処理を実行し、かつサービス許可チケット(SPT)に記録された許可範囲内のアクセス(ex. データの読み取り、書き込みなど)処理を実行する。この処理については後段で詳細に説明する。

【0083】また、図1に示すように、デバイスマネージャ200、パーティションマネージャ300の上位にコード管理機関400が設定され、個々のデバイスマネージャ、パーティションマネージャに各エンティティの識別情報としてのコードを割り振る処理を行なっている。これら各マネージャに付与されたコードは、前述のパーティション登録チケット(PRT)、ファイル登録チケット(FRT)等のアクセスコントロールチケットの格納データとされる。

【0084】デバイス100がユーザに提供(ex. 貸与、販売)されユーザが利用する以前に、提供デバイスを管理するデバイスマネージャ(DM)200が設定され、その提供デバイス内にデバイスマネージャコード他、デバイスマネージャの管理情報が書き込まれる。これらのデータ詳細については後述する。

【0085】本発明のメモリデバイスを利用したデータ管理システムにおける公開鍵証明書の発行処理と各エンティティの関係について、図2を用いて説明する。

【0086】図2は、デバイス管理エンティティとしてのデバイスマネージャ、デバイスに設定された各パーティションの管理エンティティとして、2つのパーティションマネージャ300A、300B、デバイスマネージャ200に対して識別コードを付与するコード管理機関400を示している。さらに、デバイスマネージャ200の管轄する登録局210からの公開鍵証明書発行要求に応じて、デバイスマネージャ200、デバイスマネージャ管轄の各機器(パーティション登録チケット(PRT)発行手段(PRT Issuer)210、あるいはデバイス100に対応するデバイス対応公開鍵証明書(CERT-DEV)を発行するデバイスマネージャ対応認証局(CA(DEV): Certificate Authority)610、パーティションマネージャ300A、300B管轄の各機器(ファイル登録チケット(FRT)発行手段(FRT Issuer)310、サービス許可チケット発行手段(SPT)320、チケットユーザであるデバイスアクセス機器としてのリーダライタ711~714、あるいはデバイス100のパーティションに対応するパーティション対応公開鍵証明書(CERT-PAR)を発行するパーティションマネージャ対応認証局(CA(PAR): Certificat

e Authority)620、630が存在する。

【0087】なお、図2には、認証局をデバイスマネージャ対応認証局: CA(Certificate Authority) for DM(またはCA(DEV))610と、パーティションマネージャ対応認証局: CA for PAR(またはCA(PAR))620、630と個別に有する構成を示しているが、両機能を持つ唯一の認証局を設けたり、複数のパーティションマネージャに対応する共通の認証局とデバイスマネージャ対応認証局を別々に設けたり、その構成は自由である。

【0088】デバイスマネージャ200、パーティションマネージャ300A、300Bは、自己の公開鍵証明書、各マネージャの管理する各機器(チケット発行手段、チケットユーザ)の公開鍵証明書、または、デバイス100からの公開鍵証明書発行要求を受理し、受理した発行要求の検証を行ない、検証の後、証明書発行要求を認証局に対して転送する処理を行なうとともに、発行された公開鍵証明書の管理処理を行なう登録局(RA: Registration Authority)220、330を有する。

【0089】これら登録局(RA)220、330を介して各認証局(CA)610、620、630から発行された公開鍵証明書はデバイス100に格納され、デバイス100に対する処理としての例えばパーティションの設定処理、あるいはパーティションに対する処理としての例えばファイル設定処理、さらにファイルに対するアクセス処理等の際の相互認証処理、あるいは前述した各チケットの正当性検証処理に使用される。これら公開鍵証明書の発行処理、公開鍵証明書を使用した各処理の詳細については後述する。

【0090】図2において、デバイス100は、パーティションとしてパーティションマネージャ1、300Aの管理パーティション: PM1Area、パーティションマネージャ2、300Bの管理パーティション: PM2Areaを有し、さらに、デバイスマネージャ200の管理領域としてのDMAreaを有する。

【0091】デバイスマネージャ200はパーティション登録チケット発行手段(PRT Issuer)210を有し、パーティションマネージャ300は、ファイル登録チケット発行手段(FRT Issuer)310、およびサービス許可チケット発行手段(SPT Issuer)320を有しており、それぞれ各チケットを発行する。

【0092】パーティションマネージャ1、300Aは、PRT、FRT、SPT各チケット毎に異なる専用のリーダ/ライタ(デバイスに対するデータ読み出し書き込み用のインタフェース)711~713を有した構成であり、パーティションマネージャ2、300Bは、各チケットに共通のリーダ/ライタ714を有した構成を示している。リーダ/ライタはこのように様々な構成をとることが可能である。

【0093】さらに図3を用いてエンティティの具体例

について説明する。図3には、デバイスに設定されたパーティションを利用したサービスを提供するサービス主体としてのパーティションマネージャとして東西鉄道株式会社および南北鉄道株式会社の2つのサービス主体を想定し、これらパーティションマネージャに対してパーティションの設定登録を行なうデバイスマネージャとして日本鉄道グループという組織を想定したデバイス利用構成例を示している。

【0094】東西鉄道株式会社は、ユーザのデバイスに設定された自身の管理するパーティション：PM1内に複数のファイルを登録している。すなわち、定期券用ファイル、プリペイド用ファイル、その他用ファイルである。各サービス主体としてのパーティションマネージャは自己の提供するサービスに応じて設定されたデバイスマネージャによって割り当てられたパーティション内に様々なファイルを登録できる。ただし、ファイルの設定登録にはファイル登録チケット（FRT）が必要となる。

【0095】東西鉄道株式会社は、デバイスの1つのパーティション：PM1Areaを管理するパーティションマネージャとして機能する。パーティション：PM1Areaは、デバイスマネージャとしての日本鉄道グループによって、日本鉄道グループのPRTIssuerの発行するパーティション登録チケット（PRT）に登録されたルールに従った認証、検証等の処理が実行され、かつパーティション登録チケット（PRT）に登録された許可範囲内のパーティションの設定登録処理によって設定されて、東西鉄道株式会社に付与される。

【0096】東西鉄道株式会社は、付与されたパーティション：PM1Areaに自身の提供するサービスに応じて様々なファイルを設定する。例えば定期券ファイル、プリペイド用ファイルであり、定期券ファイル内のデータ格納エリアには例えば、定期券使用者名、使用期間、利用区間など定期券管理データとして必要な各種データを記録する。また、プリペイド用ファイルには、使用者名、プリペイド金額、残額データなどが記録される。このファイル設定処理には、東西鉄道のFRTIssuerの発行するファイル登録チケット（FRT）に登録されたルールに従った認証、検証等の処理が実行され、かつファイル登録チケット（FRT）に登録された許可範囲内のファイルの設定登録処理によって設定される。

【0097】このように様々なファイルの設定されたデバイスがユーザによって使用される。例えば、ユーザがデバイスを使用してデバイスアクセス機器としてのリーダライタを備えた改札にデバイスをセットして利用することが可能である。例えば改札に備えられた正当なリーダライタにより、定期券用ファイルのアクセスが実行されて、利用区間の読み取りが行われる。またプリペイド用ファイルにアクセスして、プリペイド用ファイル内の

残金データの更新処理が実行される。

【0098】デバイス内の、いずれのファイルにアクセスしてどのような処理（読み取り、書き込み、減額etc）を実行するかは、東西鉄道のサービス許可チケット（SPT）発行手段（SPT issuer）の発行するサービス許可チケット（SPT）に登録されている。例えば改札に備えられた正当なデバイスアクセス機器としてのリーダライタにはこれらのチケットが格納されており、チケットに登録されたルールに従ってデバイス間との認証処理、チケット検証等の処理が実行される。デバイスアクセス機器としてのリーダライタおよびデバイス相互が正当な機器であり、使用チケットが正当である場合にサービス許可チケット（SPT）に登録された許可範囲内の処理（ex. ファイル内のデータ読み取り、書き込み、減額etc）が実行されることになる。

【0099】パーティション登録チケット（PRT）、ファイル登録チケット（FRT）、およびサービス許可チケット（SPT）の各種チケットを発行するチケット発行手段（Ticket Issuer）とチケット発行手段によって発行されたチケットを利用するチケットユーザ（Ticket User）の一般的な対応関係を図4に示す。

【0100】チケット発行手段（Ticket Issuer）は、図1他で説明したように、デバイスマネージャ、あるいはパーティションマネージャの管理下にあり、デバイスに対する処理に応じたパーティション登録チケット（PRT）、ファイル登録チケット（FRT）、およびサービス許可チケット（SPT）の各種チケットを発行する。チケットユーザ（Ticket User）は、チケット発行手段によって発行されたチケットを利用する機器、手段であり、具体的には例えばデバイスに対するデータ書き込み、読み取りなどの処理を実行するデバイスアクセス機器としてのリーダライタ等の機器が相当する。

【0101】図4に示すように、チケットユーザは、複数のチケットを格納して使用することが可能である。また、単一のチケット、例えば図3を用いて説明した定期券用ファイルの区間データ読み取りのみの実行を許可したサービス許可チケット（SPT）のみを格納し、区間データ読み取りのみの処理を実行する構成とすることも可能である。

【0102】例えば、あるサービス主体（パーティションマネージャ）である鉄道会社の定期券読み取り専用の改札には、上述の定期券用ファイルの区間データ読み取りのみの実行を許可したサービス許可チケット（SPT）のみを格納したデバイスアクセス機器としてのリーダライタを設定して、ユーザが所有するデバイスから区間データの読み取り処理を実行する。例えば、定期券、プリペイド双方の処理を実行する改札のデバイスアクセス機器としてのリーダライタには上述の定期券用ファイルの区間データ読み取りのみの実行を許可したサービス許可チケット（SPT）、およびプリペイド用ファイル

の残金データ減額処理を許可したサービス許可チケット（SPT）を併せて格納し、定期券用ファイル、およびプリペイド用ファイル両ファイルに対する処理を実行可能とすることも可能である。

【0103】また、パーティション登録チケット（PRT）、ファイル登録チケット（FRT）、およびサービス許可チケット（SPT）を格納し、パーティション登録、ファイル登録、ファイル内のデータアクセス等のすべての処理を実行可能としたチケットユーザ（ex. リーダライタ）を構成することも可能である。このようにチケットユーザの実行可能な処理は、チケットユーザが適用可能なチケットによって決定されることになる。

【0104】[A2. デバイスの構成] 次に、上述の複数のパーティションにデータ格納領域を分割されたメモリを持つデバイスについて説明する。図5にデバイスの構成図を示す。

【0105】図5に示すように、デバイス100は、プログラム実行機能、演算処理機能を持つCPU（Central Processing Unit）101を有し、デバイスアクセス機器としてのリーダライタ等、外部機器との通信処理用のインタフェース機能を持つ通信インタフェース102、CPU101によって実行される各種プログラム、例えば暗号処理プログラムなどを記憶したROM（Read Only Memory）103、実行プログラムのロード領域、また、各プログラム処理におけるワーク領域として機能するRAM（Random Access Memory）104、外部機器との認証処理、電子署名の生成、検証処理、格納データの暗号化、復号処理等の暗号処理を実行する暗号処理部105、前述したパーティション、ファイルが設定格納されるとともに、各種鍵データを含むデバイスの固有情報を格納した例えばEEPROM（Electrically Erasable Programmable ROM）によって構成されるメモリ部106を有する。メモリ部106（ex. EEPROM）106に格納される情報については、後段で詳述する。

【0106】メモリ部106のデータ格納構成を図6に示す。メモリ部は例えば、EEPROM（Electrically Erasable Programmable ROM）と呼ばれる電氣的に書き換え可能な不揮発性メモリの一形態であるフラッシュメモリである。

【0107】図6に示すように、本実施例においては、1ブロック32バイト、ブロック数0xFFFFのデータ格納領域を持ち、主要領域としてパーティション領域、未使用領域、デバイス固有情報およびデバイス内パーティション情報領域を持つ。

【0108】パーティション領域には、前述のパーティションマネージャによる管理領域であるパーティションが設定登録される。なお、図6に示すメモリは既にパーティションが設定された例を示しているが、新規に製造されたデバイスには、パーティションが設定されておらずパーティション領域は存在しない。パーティション

は、前述したように、デバイスマネージャの管理するパーティション登録チケット（PRT）発行手段（PRT Issuer）の発行したPRTチケットに基づいて各サービス主体としてのパーティションマネージャが所定の手続き、すなわちパーティション登録チケット（PRT）に設定されたルールに従ってデバイス内のメモリに設定する。

【0109】デバイス固有情報およびデバイス内パーティション情報領域には、デバイス製造エンティティの情報、デバイスマネージャに関する情報、設定パーティション情報、デバイスに対するアクセスを実行してパーティションの設定登録処理を実行する際に必要となる鍵情報などが格納される。これら格納情報の詳細については後述する。なお、デバイス固有情報領域の格納データは、後述する相互認証時に適用するデバイスの固有値としてのIDmに対応するデータとして使用可能である。

【0110】また、図に示すようにパーティション領域は、さらに1以上のファイル領域、未使用領域、パーティション固有情報およびパーティション内ファイル領域を有する。ファイル領域は、パーティションマネージャであるサービス主体が、例えば前述したような定期券用、プリペイド用などサービス毎に設定したファイルを格納する領域である。未使用領域は、さらにファイル設定可能な領域である。パーティション固有情報およびパーティション内ファイル情報領域は、例えばパーティション内のファイルに関する情報、ファイルアクセス処理に必要な鍵情報などが格納される。これら格納情報の詳細については後述する。

【0111】[A3. デバイスマネージャの構成] 次に、デバイスマネージャの構成について図7を用いて説明する。デバイスマネージャは、ユーザに提供（販売または貸与）されるデバイスの管理エンティティである。

【0112】デバイスマネージャ200は、デバイス内のメモリ部の分割領域として設定されるパーティションを利用してサービスを提供するサービス主体としてのパーティションマネージャからの要求に応じてデバイスに対するパーティション設定を可能化するパーティション登録チケット（PRT）を発行するパーティション登録チケット（PRT）発行手段（PRT Issuer）210を有する。

【0113】さらに、デバイスマネージャ200は、デバイスに対応するデバイス対応公開鍵証明書（CERT-DEV）を発行する。デバイスマネージャ200は、デバイスからの公開鍵証明書発行要求を受理し、受理した発行要求の検証を行ない、検証の後、証明書発行要求を認証局（CA（DEV）：Certificate Authority）610に対して転送する処理を行なうとともに、発行された公開鍵証明書の管理処理を行なう登録局（RA：Registration Authority）220としての機能を有する。

【0114】図7に示すように、デバイスマネージャ2

00のパーティション登録チケット(PRT)発行手段(PRT Issuer)210は、制御手段211と、データベース212を有し、データベース212は、パーティション登録チケット(PRT)の発行管理データとして、チケットの発行管理用のデータ、例えば、チケット発行先のパーティションマネージャ識別子、チケット識別子、チケットユーザ(e.g. リードライタ、PC、etc)識別子などを対応付けたデータが格納される。

【0115】また、登録局(RA:Registration Authority)220は、制御部221、公開鍵証明書の発行管理用のデータベース222を有し、公開鍵証明書の発行管理データとして、例えば公開鍵証明書を発行したデバイス識別子、公開鍵証明書の識別子(シリアルナンバ)等を対応付けたデータが格納される。

【0116】デバイスマネージャ200のパーティション登録チケット(PRT)発行手段(PRT Issuer)210の制御手段211は、パーティションマネージャとのデータ通信により、パーティション登録チケット(PRT)の発行処理を実行する。また、登録局(RA:Registration Authority)220の制御手段221は、デバイスに対する公開鍵証明書の発行処理を実行し、この際、デバイスとの通信、デバイスマネージャ対応認証局(CA(DEV))610との通信を実行する。これらの処理の詳細については後段で説明する。ここでは、制御手段211、221の構成について図8を用いて説明する。

【0117】制御手段211、221はいずれも例えばデータ処理手段としてのPCと同様の構成により実現され、具体的には例えば図8に示すような構成を持つ。制御手段の構成について説明する。制御部211は各種処理プログラムを実行する中央演算処理装置(CPU:Central Processing Unit)によって構成される。ROM(Read only Memory)2112は、暗号処理プログラム等の実行処理プログラムを記憶したメモリである。RAM(Random Access Memory)2113は、制御部211が実行するプログラム、例えばデータベース管理プログラム、暗号処理プログラム、通信プログラム等、実行プログラムの格納領域、またこれら各プログラム処理におけるワークエリアとして使用される。

【0118】表示部2114は、液晶表示装置、CRTなどの表示手段を有し、制御部211の制御の下、様々なプログラム実行時のデータ、例えば処理対象のデータ内容等を表示する。入力部2115は、キーボードや、例えばマウス等のポインティングデバイスを有し、これら各入力デバイスからのコマンド、データ入力を制御部211に出力する。HDD(Hard Disk Drive)2116は、データベース管理プログラム、暗号処理プログラム、通信プログラム等のプログラム、さらに各種データが格納される。

【0119】ドライブ2117は、例えばHD(Hard D

isk)や、FD(Floppy Disk)等の磁気ディスク、CD-ROM(Compact Disk ROM)などの光ディスク、ミニディスク等の光磁気ディスク、ROMやフラッシュメモリなどの半導体メモリ等の各種記録媒体に対するアクセスを制御する機能を持つ。磁気ディスク等の各種記録媒体はプログラム、データ等を記憶する。通信インタフェース2118は、ネットワーク、ケーブル接続、電話回線等の有線、無線を介した通信のインタフェースとして機能し、ユーザのデバイス、パーティションマネージャ、認証局等の各エンティティとの通信インタフェースとして機能する。

【0120】[A4. パーティションマネージャの構成]次に、パーティションマネージャの構成について図9を用いて説明する。パーティションマネージャは、ユーザに提供(販売または貸与)されるデバイスに設定されたパーティションの管理エンティティである。

【0121】パーティションマネージャ300は、デバイスマネージャから付与されたパーティション登録チケット(PRT)を用いて、付与されたPRTに記録されたルールに従って、ユーザのデバイス内のメモリ部に分割領域としてパーティションを設定し、設定されたパーティションを利用したサービスを提供する。

【0122】設定されたパーティションにはサービス、データに応じたファイルを設定することが可能である。ただし、ファイル設定処理には、ファイル登録チケット(FRT)を取得することが必要であり、ファイル内のデータの読み出し、書き込みなどのデータアクセスにはサービス許可チケット(SPT)を取得することが必要である。ファイル設定、データアクセス処理はチケットユーザ、すなわち具体的には、例えば専用のデバイスアクセス機器としてのリードライタなどがチケットを使用して実行する。

【0123】パーティションマネージャ300は、このようなチケットユーザに対するチケット発行処理手段としてのファイル登録チケット(FRT)発行手段(FRT Issuer)310、およびサービス許可チケット(SPT)発行手段(SPT Issuer)320を有する。

【0124】さらに、パーティションマネージャ300は、デバイスの各パーティションに対応するパーティション対応公開鍵証明書(CERT-PAR)を発行する。パーティションマネージャ300は、デバイスからの公開鍵証明書発行要求を受理し、受理した発行要求の検証を行ない、検証の後、証明書発行要求を認証局(CA(PAR):Certificate Authority)620に対して転送する処理を行なうとともに、発行された公開鍵証明書の管理処理を行なう登録局(RA:Registration Authority)330としての機能を有する。

【0125】図9に示すように、パーティションマネージャ300のファイル登録チケット(FRT)発行手段(FRT Issuer)310は、制御手段311と、デー

データベース312を有し、データベース312は、ファイル登録チケット(FRT)の発行管理データとして、チケットの発行管理用のデータ、例えば、チケット発行先のチケットユーザ(ex. リーダライタ、PC, etc)識別子、チケット識別子などを対応付けたデータを格納する。

【0126】さらにパーティションマネージャ300のサービス許可チケット(SPT)発行手段(SPT Issuer)320は、制御手段321と、データベース322を有し、データベース322は、サービス許可チケット(SPT)の発行管理データとして、チケットの発行管理用のデータ、例えば、チケット発行先のチケットユーザ(ex. デバイスアクセス機器としてのリーダライタ、PC, etc)識別子、チケット識別子などを対応付けたデータを格納する。

【0127】また、登録局(RA:Registration Authority)330は、公開鍵証明書の発行管理用のデータベース332を有し、公開鍵証明書の発行管理データとして、例えば公開鍵証明書を発行したデバイス識別子、パーティション識別子、公開鍵証明書の識別子(シリアルナンバ)等を対応付けたデータが格納される。

【0128】パーティションマネージャ300のファイル登録チケット(FRT)発行手段(FRT Issuer)310の制御手段311は、チケットユーザ(ex. デバイスアクセス機器としてのリーダライタ、PC, etc)とのデータ通信により、ファイル登録チケット(FRT)の発行処理を実行し、サービス許可チケット(SPT)発行手段(Ticket Issuer)320の制御手段321は、チケットユーザ(ex. デバイスアクセス機器としてのリーダライタ、PC, etc)とのデータ通信により、サービス許可チケット(SPT)の発行処理を実行する。また、登録局(RA:Registration Authority)330の制御手段331は、デバイスに対する公開鍵証明書の発行処理を実行し、この際、デバイスとの通信、パーティションマネージャ対応認証局(CA(PAR))620との通信を実行する。これらの処理の詳細については後段で説明する。

【0129】なお、パーティションマネージャ300の制御手段311、321、331の構成は、図8を用いて説明した前述のデバイスマネージャにおける制御手段と同様の構成であるので説明を省略する。

【0130】[A5. チケットユーザ(デバイスアクセス機器としてのリーダライタ)の構成] デバイスアクセス機器としてのリーダライタはデバイスに対するパーティションの設定、ファイルの設定、データの読み取り、書き込み、金額データの減算、加算などの様々な処理を実行する機器として構成される。デバイスに対する処理は、処理の際に適用するパーティション登録チケット(PRT)、ファイル登録チケット(FRT)、またはサービス許可チケット(SPT)に記録されたルールに

従う。すなわち、デバイスに対するすべての処理はこれら適用するチケットによって制限される。

【0131】デバイスアクセス機器としてのリーダライタの構成例を図10に示す。図10に示すように、リーダライタ700は、プログラム実行機能、演算処理機能を持つCPU(Central Processing Unit)701を有し、デバイス、チケット発行手段(Ticket Issuer)等、外部機器との通信処理用のインタフェース機能を持つ通信インタフェース702、CPU701によって実行される各種プログラム、例えば暗号処理プログラムなどを記憶したROM(Read Only Memory)703、実行プログラムのロード領域、また、各プログラム処理におけるワーク領域として機能するRAM(Random Access Memory)704、外部機器との認証処理、電子署名の生成、検証処理、格納データの暗号化、復号処理等の暗号処理を実行する暗号処理部705、認証処理、暗号化、復号処理用の各種鍵データ、およびリーダライタの固有情報を格納した例えばEEPROM(Electrically Erasable Programmable ROM)によって構成されるメモリ部706を有する。

【0132】[A6. 公開鍵証明書] 本発明のパーティション分割メモリ領域を持つデバイスの利用において、各エンティティ、チケット発行手段、チケットユーザ、デバイス等の相互間におけるデータ通信においては、データ送信側とデータ受信側とが互いに正規なデータ送受信対象であることを確認した上で、必要な情報を転送する、データ転送の際のセキュリティ構成を実現する手法として、転送データの暗号化処理、データに対する署名生成、検証処理が適用される。

【0133】暗号化データは、所定の手続きによる復号化処理によって利用可能な復号データ(平文)に戻すことができる。このような情報の暗号化処理に暗号化鍵を用い、復号化処理に復号化鍵を用いるデータ暗号化、復号化方法は従来からよく知られている。

【0134】暗号化鍵と復号化鍵を用いるデータ暗号化・復号化方法の態様には様々な種類があるが、その1つの例としていわゆる公開鍵暗号方式と呼ばれる方式がある。公開鍵暗号方式は、発信者と受信者の鍵を異なるものとして、一方の鍵を不特定のユーザが使用可能な公開鍵として、他方を秘密に保つ秘密鍵とするものである。例えば、データ暗号化鍵を公開鍵とし、復号鍵を秘密鍵とする。あるいは、認証子生成鍵を秘密鍵とし、認証子検証鍵を公開鍵とする等の態様において使用される。

【0135】暗号化、復号化に共通の鍵を用いるいわゆる共通鍵暗号化方式と異なり、公開鍵暗号方式では秘密に保つ必要のある秘密鍵は、特定の1ユーザが持てばよい鍵の管理において有利である。ただし、公開鍵暗号方式は共通鍵暗号化方式に比較してデータ処理速度が遅く、秘密鍵の配送、ディジタル署名等のデータ量の少ない対象に多く用いられている。公開鍵暗号方式の代表

的なものにはRSA (Rivest-Shamir-Adleman) 暗号がある。これは非常に大きな2つの素数(例えば150桁)の積を用いるものであり、大きな2つの素数(例えば150桁)の積の素因数分解する処理の困難さを利用している。

【0136】公開鍵暗号方式では、不特定多数に公開鍵を使用可能とする構成であり、配布する公開鍵が正当なものであるか否かを証明する証明書、いわゆる公開鍵証明書を使用する方法が多く用いられている。例えば、利用者Aが公開鍵、秘密鍵のペアを生成して、生成した公開鍵を認証局に対して送付して公開鍵証明書を認証局から入手する。利用者Aは公開鍵証明書を一般に公開する。不特定のユーザは公開鍵証明書から所定の手続きを経て公開鍵を入手して文書等を暗号化して利用者Aに送付する。利用者Aは秘密鍵を用いて暗号化文書等を復号する等のシステムである。また、利用者Aは、秘密鍵を用いて文書等に署名を付け、不特定のユーザが公開鍵証明書から所定の手続きを経て公開鍵を入手して、その署名の検証を行なうシステムである。

【0137】公開鍵証明書は、公開鍵暗号方式における認証局(CA:Certificate Authority)が発行する証明書であり、ユーザが自己のID、公開鍵等を認証局に提出することにより、認証局側が認証局のIDや有効期限等の情報を付加し、さらに認証局による署名を付加して作成される証明書である。

【0138】図11に公開鍵証明書のフォーマットの概略を示す。各データの概要について説明する。証明書のバージョン(version)番号は、公開鍵証明書フォーマットのバージョンを示す。証明書の通し番号は、シリアルナンバ(SN:Serial Number)であり、公開鍵証明書発行局(認証局:CA)によって設定される公開鍵証明書のシリアルナンバである。署名アルゴリズム識別子フィールド(Signature algorithm Identifier)の署名アルゴリズム(algorithm)、アルゴリズムパラメータ(parameters)は、公開鍵証明書の署名アルゴリズムとそのパラメータを記録するフィールドである。なお、署名アルゴリズムとしては、楕円曲線暗号およびRSAがあり、楕円曲線暗号が適用されている場合はパラメータおよび鍵長が記録され、RSAが適用されている場合には鍵長が記録される。発行局(認証局:CA)の名前は、公開鍵証明書の発行者、すなわち公開鍵証明書発行局(CA)の名称(Issuer)が識別可能な形式(Distinguished Name)で記録されるフィールドである。証明書の有効期限(validity)は、証明書の有効期限である開始日時、終了日時が記録される。公開鍵証明書の利用者名(Subject)は、ユーザである認証対象者の識別データが記録される。具体的には例えばユーザ機器のIDや、サービス提供主体のID等の識別子またはカテゴリが記録される。利用者公開鍵フィールド(subject Public Key Info)の鍵アルゴリズム(algorithm)と鍵(subjec

t Public key)は、ユーザの公開鍵情報としての鍵アルゴリズム、鍵情報そのものを格納するフィールドである。オプション領域には、ユーザの属性データ、その他公開鍵証明書の発行、利用に伴うオプションデータを記録する。属性データとしては、ユーザの所属グループ情報としてのデバイスマネージャコード(DMC)、パーティションマネージャコード(PMC)が記録される。なお、ここでユーザは公開鍵証明書のユーザであり、例えばデバイスマネージャ、パーティションマネージャ、チケットユーザ、チケット発行手段、デバイスなどである。

【0139】オプション領域には、さらにカテゴリ情報として、チケットユーザ、チケット発行手段、デバイス、デバイスマネージャ、パーティションマネージャなどのエンティティ、機器種別を示すカテゴリが記録される。

【0140】なお、デバイスマネージャがパーティション登録チケット発行手段(PRT Issuer)を兼ねる場合は、後述するパーティション登録チケット発行手段コード(PRTIC:PRT Issuer Code)は、デバイスマネージャコード(DMC)として設定可能であり、また、パーティションマネージャがファイル登録チケット発行手段、サービス許可チケット発行手段を兼ねる場合は、ファイル登録チケット発行手段コード(FRTIC:FRT Issuer Code)、サービス許可チケット発行手段コード(SPTIC:SPT Issuer Code)をパーティションマネージャコード(PMC)として設定可能である。なお、これらのコードは後述する各チケット(PRT, FRT, SPTなど)にも記録される。

【0141】また、各チケット発行手段にデバイスマネージャコード(DMC)、パーティションマネージャコード(PMC)と異なる独自のコードを割り当てる構成としてもよい。この場合のコード付与は、前述のコード管理機関が実行する。

【0142】発行局署名は、公開鍵証明書発行局(CA)の秘密鍵を用いて公開鍵証明書のデータに対して実行される電子署名であり、公開鍵証明書の利用者は、公開鍵証明書発行局(CA)の公開鍵を用いて検証を行ない、公開鍵証明書の改竄有無がチェック可能となっている。

【0143】公開鍵暗号方式を用いた電子署名の生成方法について、図12を用いて説明する。図12に示す処理は、ECDSA(Elliptic Curve Digital Signature Algorithm)、IEEE P1363/D3を用いた電子署名データの生成処理フローである。なお、ここでは公開鍵暗号として楕円曲線暗号(Elliptic Curve Cryptography(以下、ECCと呼ぶ))を用いた例を説明する。なお、本発明のデータ処理装置においては、楕円曲線暗号以外にも、同様の公開鍵暗号方式における、例えばRSA暗号(Rivest, Shamir, Adleman)など(ANSI X9.3

1) ) を用いることも可能である。

【0144】図12の各ステップについて説明する。ステップS1において、 $p$ を標数、 $a$ 、 $b$ を楕円曲線の係数（楕円曲線： $y^2 = x^3 + ax + b$ ,  $4a^3 + 27b^2 \neq 0 \pmod{p}$ ）、 $G$ を楕円曲線上のベースポイント、 $r$ を $G$ の位数、 $K_s$ を秘密鍵（ $0 < K_s < r$ ）とする。ステップS2において、メッセージ $M$ のハッシュ値を計算し、 $f = \text{Hash}(M)$ とする。

【0145】ここで、ハッシュ関数を用いてハッシュ値を求める方法を説明する。ハッシュ関数とは、メッセージを入力とし、これを所定のビット長のデータに圧縮し、ハッシュ値として出力する関数である。ハッシュ関数は、ハッシュ値（出力）から入力を予測することが難しく、ハッシュ関数に入力されたデータの1ビットが変化したとき、ハッシュ値の多くのビットが変化し、また、同一のハッシュ値を持つ異なる入力データを探し出すことが困難である特徴を有する。ハッシュ関数としては、MD4、MD5、SHA-1などが用いられる場合もあるし、DES-CBCが用いられる場合もある。この場合は、最終出力値となるMAC（チェック値：ICVに相当する）がハッシュ値となる。

【0146】続けて、ステップS3で、乱数 $u$ （ $0 < u < r$ ）を生成し、ステップS4でベースポイントを $u$ 倍した座標 $V(X_v, Y_v)$ を計算する。なお、楕円曲線上の加算、2倍算は次のように定義されている。

【0147】

【数1】 $P=(X_a, Y_a), Q=(X_b, Y_b), R=(X_c, Y_c)=P+Q$ とすると、 $P \neq Q$ の時（加算）、

$$X_c = \lambda^2 - X_a - X_b$$

$$Y_c = \lambda \times (X_a - X_c) - Y_a$$

$$\lambda = (Y_b - Y_a) / (X_b - X_a)$$

$P=Q$ の時（2倍算）、

$$X_c = \lambda^2 - 2X_a$$

$$Y_c = \lambda \times (X_a - X_c) - Y_a$$

$$\lambda = (3(X_a)^2 + a) / (2Y_a)$$

【0148】これらを用いて点 $G$ の $u$ 倍を計算する（速度は遅いが、最もわかりやすい演算方法として次のように行う。 $G, 2 \times G, 4 \times G, \dots$ を計算し、 $u$ を2進数展開して1が立っているところに対応する $2^i \times G$ （ $G$ を $i$ 回2倍算した値（ $i$ は $u$ のLSBから数えた時のビット位置））を加算する。

【0149】ステップS5で、 $c = X_v \bmod r$ を計算し、ステップS6でこの値が0になるかどうか判定し、0でなければステップS7で $d = [(f + cK_s) / u] \bmod r$ を計算し、ステップS8で $d$ が0であるかどうか判定し、 $d$ が0でなければ、ステップS9で $c$ および $d$ を電子署名データとして出力する。仮に、 $r$ を160ビット長の長さであると仮定すると、電子署名データは320ビット長となる。

【0150】ステップS6において、 $c$ が0であった場

合、ステップS3に戻って新たな乱数を生成し直す。同様に、ステップS8で $d$ が0であった場合も、ステップS3に戻って乱数を生成し直す。

【0151】次に、公開鍵暗号方式を用いた電子署名の検証方法を、図13を用いて説明する。ステップS11で、 $M$ をメッセージ、 $p$ を標数、 $a$ 、 $b$ を楕円曲線の係数（楕円曲線： $y^2 = x^3 + ax + b$ ,  $4a^3 + 27b^2 \neq 0 \pmod{p}$ ）、 $G$ を楕円曲線上のベースポイント、 $r$ を $G$ の位数、 $G$ および $K_s \times G$ を公開鍵（ $0 < K_s < r$ ）とする。ステップS12で電子署名データ $c$ および $d$ が $0 < c < r$ 、 $0 < d < r$ を満たすか検証する。これを満たしていた場合、ステップS13で、メッセージ $M$ のハッシュ値を計算し、 $f = \text{Hash}(M)$ とする。次に、ステップS14で $h = 1/d \bmod r$ を計算し、ステップS15で $h_1 = fh \bmod r$ 、 $h_2 = ch \bmod r$ を計算する。

【0152】ステップS16において、既に計算した $h_1$ および $h_2$ を用い、点 $P = (X_p, Y_p) = h_1 \times G + h_2 \cdot K_s \times G$ を計算する。電子署名検証者は、ベースポイント $G$ および $K_s \times G$ を知っているため、図12のステップS4と同様に楕円曲線上の点のスカラー倍の計算ができる。そして、ステップS17で点 $P$ が無限遠点かどうか判定し、無限遠点でなければステップS18に進む（実際には、無限遠点の判定はステップS16でできてしまう。つまり、 $P = (X, Y)$ 、 $Q = (X, -Y)$ の加算を行うと、 $\lambda$ が計算できず、 $P + Q$ が無限遠点であることが判明している）。ステップS18で $X_p \bmod r$ を計算し、電子署名データ $c$ と比較する。最後に、この値が一致していた場合、ステップS19に進み、電子署名が正しいと判定する。

【0153】電子署名が正しいと判定された場合、データは改竄されておらず、公開鍵に対応した秘密鍵を保持する者が電子署名を生成したことがわかる。

【0154】ステップS12において、電子署名データ $c$ または $d$ が、 $0 < c < r$ 、 $0 < d < r$ を満たさなかった場合、ステップS20に進む。また、ステップS17において、点 $P$ が無限遠点であった場合もステップS20に進む。さらにまた、ステップS18において、 $X_p \bmod r$ の値が、電子署名データ $c$ と一致していなかった場合にもステップS20に進む。

【0155】ステップS20において、電子署名が正しくないと判定された場合、データは改竄されているか、公開鍵に対応した秘密鍵を保持する者が電子署名を生成したのではないことがわかる。

【0156】本発明のシステムにおけるデバイスは、デバイスマネージャの管理登録局を介してデバイスに対して発行されるデバイス対応の公開鍵証明書（CERT-DEV）をデバイスに格納し、さらに、パーティションマネージャの管理登録局を介してデバイスのパーティションに対して発行されるパーティション対応の公開鍵証明書（CE



RT-PAR) をデバイスの各パーティションに格納する。これらの公開鍵証明書は、デバイスに対する処理、すなわちパーティション登録チケット (PRT) を適用したパーティション登録設定処理、ファイル登録チケット (FRT) を適用したファイル登録設定処理、さらにサービス許可チケット (SPT) を適用したデータ処理において、チケットユーザ (ex. デバイスアクセス機器としてのリーダーライター) とデバイス間の相互認証、署名生成、検証処理等に適用される。これらの処理の具体例については、後段でフローを用いて説明する。

【0157】 [A7. デバイスのメモリ部における格納データ] 次に、本発明のパーティション分割されたメモリ領域を持つデバイスの格納データについて説明する。先に、図6を用いて説明した通り、デバイスは例えばEEPROMからなるメモリ部を持ち、主要領域としてパーティション領域、未使用領域、デバイス固有情報およびデバイス内パーティション情報領域を有する。これら各領域の格納データについて以下、図を参照して順次説明する。

【0158】 (A7. 1. デバイス固有情報およびデバイス内パーティション情報領域) まず、デバイス固有情報およびデバイス内パーティション情報領域の各データについて説明する。デバイス固有情報およびデバイス内パーティション情報領域には、デバイス製造エンティティの情報、デバイスマネージャに関する情報、設定パーティション情報、デバイスに対するアクセスを実行してパーティションの設定登録処理を実行する際に必要となる鍵情報などが格納される。

【0159】 図14は、製造情報ブロック (Manufacture Information Block) のデータ構成を示している。各領域の数値は、バイト数を示す。図6を用いて説明したように、本実施例の構成では1ブロック: 32バイト構成である。なお、図中グレー部分は暗号化されたデータでも、暗号化されていなくてもよい。

【0160】 製造情報ブロック (Manufacture Information Block) には、以下のデータが格納される。

- \* Writable Flag : ブロックへ書き込みが許可されているかの判別フラグ (ex. 0xffff : 書込許可, others : 書込不可)
- \* Manufacture Code : カード製造工場識別番号
- \* Manufacture Equipment Code : カード製造ライン番号
- \* Manufacture Date : カード製造日。例えば、2001年1月1日を0x0000とする
- \* Manufacture Serial Number : カード製造シリアル番号
- \* Device Vender Code : ICチップ供給会社番号
- \* Device Code : ICチップの型番
- \* Device Parameter : その他のパラメータ

【0161】 これらのブロックに書かれる情報は一意に

なるので、これらの情報を基にデバイス (Device) 固有識別子としてIDmと定義する。なお、デバイス (Device) 固有識別子は製造情報ブロック (Manufacture Information Block) に書き込まれた情報全体、あるいは書き込まれた情報の一部、または書き込まれた情報に基づいて取得される演算データから取得する構成とすることも可能である。

【0162】 図15は、デバイス管理情報ブロック (Device Management Information Block) のデータ構成を示す。デバイス管理情報ブロック (Device Management Information Block) には以下のデータが格納される。

【0163】 \* Writable Flag : ブロックへ書き込みが許可されているかの判別フラグ (ex. 0xffff : 書込許可, others : 書込不可)

- \* DMC (Device Manager Code) : デバイスマネージャ (DM : Device Manager) の識別番号
- \* DMC Version : デバイスマネージャコード (DMC) のバージョン。例えば、DMCを更新する時の比較条件として用いられる。
- \* Total Block Number in Device : デバイス (Device) 内の全ブロック数
- \* Free Block Number in Device : デバイス (Device) 内の空きブロック数
- \* Partition Number : 現在登録されているパーティション (Partition) 数
- \* Pointer of Free Area : 空き領域のポインタ

【0164】 図16は、公開鍵系デバイス鍵定義ブロック (Device Key Definition Block (PUB)) のデータ構成を示す。公開鍵系デバイス鍵定義ブロック (Device Key Definition Block (PUB)) には以下のデータが格納される。

- 【0165】 \* PUB\_CA (DEV) Pointer : デバイスマネージャの管轄する登録局を介して公開鍵証明書の発行を行なうデバイスマネージャ対応認証局 (CA (DEV)) の公開鍵が格納されているブロックへのポインタ
- \* PUB\_CA (DEV) Size : 認証局CA (DEV) の公開鍵のサイズ
- \* PRI\_DEV Pointer : デバイス (Device) の秘密鍵が格納されているブロックへのポインタ
- \* PRI\_DEV Size : デバイス (Device) の秘密鍵のサイズ
- \* PARAM\_DEV Pointer : デバイス (Device) の公開鍵パラメータが格納されているブロックへのポインタ
- \* PARAM\_DEV Size : デバイス (Device) の公開鍵パラメータのサイズ
- \* CERT\_DEV Pointer : 認証局CA (DEV) が発行したデバイス (Device) の公開鍵証明書が格納されているブロックへのポインタ
- \* CERT\_DEV Size : 認証局CA (DEV) が発行したデバイス (Device) の公開鍵証明書のサイズ
- \* CRL\_DEV Pointer : デバイス (Device) のリボケーシ

ョンリスト (Revocation List) が格納されているブロックへのポインタ

- \* CRL\_DEV Size : デバイス (Device) のリボケーションリスト (Revocation List) のサイズ
- \* PRTIC (PRT Issuer Category) : パーティション登録チケット (PRT) 発行者カテゴリ
- \* PRTIC Version : パーティション登録チケット (PRT) 発行者カテゴリ (PRTIC) のバージョン
- \* DUTIC\_DEV (DUT Issuer Category) : データアップデートチケット (DUT : Data Update Ticket) 発行者カテゴリ
- \* DUTIC\_DEV Version : データアップデートチケット (DUT : Data Update Ticket) 発行者 (DUTIC) のバージョン

【0166】なお、上記のデータ中のリボケーションリストとは、不正デバイスのリストとして、例えばデバイス流通システムの管理者が発行するデバイス排除用リストであり、不正デバイスの識別データをリスト化したデータである。デバイスアクセス機器としてのリーダライタにセットされたデバイスがリボケーションリストに記載されたデバイスである場合は処理を停止するなどの措置をとる。

【0167】例えばデバイスに対する処理を実行するすべてのデバイスアクセス機器としてのリーダライタに常に最新のリボケーションリストを配布してデバイスに対して処理を実行する際にリストを参照して処理の実行または停止を判定する。あるいはデバイスアクセス機器としてのリーダライタの通信機能によりネットワークを介して最新のリボケーションリストを閲覧することでリストに記載された不正デバイス情報を取得して処理の実行または停止を判定する。リボケーションリストを利用した具体的処理については、フローを用いた説明中で後述する。

【0168】また、上記のデータ中のデータアップデートチケット (DUT : Data Update Ticket) は、デバイスに格納された様々なデータの更新処理を実行する際に更新処理を許可し制限するためのアクセス制限チケットであり、前述のPRT、FRT、SPTの各チケットと同様、デバイスに対するアクセスルールを記録したチケットである。このデータアップデートチケット (DUT : Data Update Ticket) については、後段でさらに詳細に説明する。

【0169】図17は、共通鍵系デバイス鍵定義ブロック (Device Key Definition Block(Common)) のデータ構成を示す。共通鍵系デバイス鍵定義ブロック (Device Key Definition Block(Common)) には以下のデータが格納される。

- 【0170】\* Mkauth\_DEV\_A Pointer : 双方向個別鍵認証用マスター鍵(MKauth\_DEV\_A)のポインタ
- \* Mkauth\_DEV\_A Size : 双方向個別鍵認証用マスター鍵

(MKauth\_DEV\_A)のサイズ

- \* Kauth\_DEV\_B Pointer : 双方向個別鍵認証用鍵(Kauth\_DEV\_B)のポインタ
- \* Kauth\_DEV\_B Size : 双方向個別鍵認証用鍵(Kauth\_DEV\_B)のサイズ
- \* Kprt Pointer : パーティション登録チケット (PRT) のMAC検証用鍵(Kprt)が格納されているブロックへのポインタ
- \* Kprt Size : パーティション登録チケット (PRT) のMAC検証用鍵(Kprt)のサイズ
- \* Kdut\_DEV1-4 Pointer : データアップデートチケット (DUT) のMAC検証用鍵(Kdut)が格納されているブロックへのポインタ
- \* Kdut\_DEV1-4 Size : データアップデートチケット (DUT) のMAC検証用鍵(Kdut)のサイズ
- \* IRL\_DEV Pointer : デバイス (Device) のリボケーションリスト (Revocation List) として、不正デバイスのデバイスID (Device ID) が格納されているブロックへのポインタ
- \* IRL\_DEV Size : デバイス (Device) のリボケーションリスト (Revocation List) のサイズ

【0171】上述のデータ中に示される双方向個別鍵認証の方法、MAC (Message Authenticate Code) 検証処理については、後段で詳細に説明する。また、Kdut\_DEVは4種類存在し、(Kdut\_DEV1, Kdut\_DEV2), (Kdut\_DEV3, Kdut\_DEV4) のペアで使われる。例えば、Kdut\_DEV1, 3はMAC生成用、Kdut\_DEV2, 4は暗号用に使われる。

【0172】図18は、デバイス鍵領域 (Device Key Area) のデータ構成を示す。デバイス鍵領域 (Device Key Area) には以下のデータが格納される。なお、デバイス鍵領域 (Device Key Area) の各格納鍵には、バージョン情報が併せて格納される。鍵の更新時には、バージョンについても併せて更新される。

- 【0173】\* IRL\_DEV : 排除デバイス (Device) 、排除機器 (デバイスアクセス機器としてのリーダライタ、PC等のチケットユーザ、チケット発行手段) の識別子 (ID) を登録したリボケーションリスト (Revocation List (Device ID))
- \* CRL\_DEV : 排除デバイス (Device) 、排除機器 (デバイスアクセス機器としてのリーダライタ、PC等のチケットユーザ、チケット発行手段) の公開鍵証明書識別子 (ex. シリアルナンバ : SN) を登録したリボケーションリスト (Revocation List (Certificate))
- \* Kdut\_DEV1 : データアップデートチケット (DUT) のMAC検証用鍵
- \* Kdut\_DEV2 : データ更新用暗号鍵
- \* Kdut\_DEV3 : データアップデートチケット (DUT) のMAC検証用鍵
- \* Kdut\_DEV4 : データ更新用暗号鍵
- \* Kprt : パーティション登録チケット (PRT) のMA

## C 検証用鍵

\* CERT\_DEV : デバイスマネージャ対応公開鍵を発行する認証局 CA (DEV) が発行したデバイス (Device) の公開鍵証明書

\* PRI\_DEV : デバイス (Device) の秘密鍵

\* PARAM\_DEV : デバイス (Device) の公開鍵パラメータ

\* PUB\_CA(DEV) : デバイスマネージャ対応公開鍵を発行する認証局 CA (DEV) の公開鍵

\* Kauth\_DEV\_B : 双方向個別鍵認証用共通鍵

\* MKauth\_DEV\_A : 双方向個別鍵認証用マスター鍵

【0174】なお、図に示すデバイス鍵領域 (Device Key Area) には Kauth\_DEV\_A : 双方向個別鍵認証用共通鍵、MKauth\_DEV\_B : 双方向個別鍵認証用マスター鍵が格納されているが、これらの鍵は、デバイスが共通鍵認証処理を行なう要請が無い場合は格納しない構成としてもよく、また、Kprt : パーティション登録チケット (PRT) の MAC 検証用鍵についても、デバイスがチケット検証処理を実行しない構成の場合には格納しない構成としてもよい。

【0175】また、IRL\_DEV : 排除デバイス (Device) のデバイス識別子 (ID) を登録したリボケーションリスト (Revocation List (Device ID))、CRL\_DEV : 排除デバイス (Device) の公開鍵証明書識別子 (ex. シリアルナンバ : SN) を登録したリボケーションリスト (Revocation List (Certificate)、についても、リボーク (排除) されたデバイスが存在しない場合、あるいは他のソースを使用してリボケーションリストを取得する構成とする場合には、リボケーションリストを格納しない構成としてもよい。

【0176】図19は、パーティション定義ブロック (Partition Definition Block) のデータ構成を示す。パーティション定義ブロック (Partition Definition Block) には以下のデータが格納される。

【0177】\* PMC (Partition Manager Code) : パーティションマネージャ (Partition Manager) に割り当てられたコード (PMC)。例えば番号。

\* PMC Version : パーティションマネージャコード (PMC) のバージョン

\* Partition Start Position : パーティション (Partition) 格納先スタートアドレス

\* Partition Size : パーティション (Partition) のサイズ

【0178】以上が、デバイスのメモリ部のデバイス固有情報およびデバイス内パーティション情報領域の各データである。

【0179】(A7. 2. パーティション領域) 次に、パーティション領域の各データについて説明する。パーティション領域は、パーティションマネージャの管理領域である。前述したように、デバイスマネージャの管理するパーティション登録チケット (PRT) 発行手段

(PRT Issuer) の発行した PRT チケットに基づいて各サービス主体としてのパーティションマネージャが所定の手続き、すなわちパーティション登録チケット

(PRT) に設定されたルールに従ってデバイス内のメモリに設定する。以下、パーティション領域のデータ構成について説明する。

【0180】図20は、パーティション管理情報ブロック (Partition Management Information Block) のデータ構成を示す。パーティション管理情報ブロック (Partition Management Information Block) には以下のデータが格納される。

【0181】\* PMC (Partition Manager Code) : パーティション (Partition) 保有者の番号

\* PMC Version : パーティションマネージャコード (PMC) のバージョン

\* Total Block Number in Partition : パーティション (Partition) 内の全ブロック数

\* Free Block Number in Partition : パーティション (Partition) 内の空きブロック数

\* Pointer of Free Area : パーティション (Partition) 内の未使用領域のポインタ

\* File Number : パーティションに現在登録されているファイル (File) 数

【0182】図21は、公開鍵系パーティション鍵情報ブロック (Partition Key Definition Block (PUB)) のデータ構成を示す。公開鍵系パーティション鍵情報ブロック (Partition Key Definition Block (PUB)) には以下のデータが格納される。

【0183】\* PUB\_CA(PAR) Pointer : パーティションマネージャの管轄登録局を介して公開鍵証明書を発行する認証局 CA (PAR) の公開鍵が格納されているブロックへのポインタ

\* PUB\_CA(PAR) Size : 認証局 CA (PAR) の公開鍵のサイズ

\* PRI\_PAR Pointer : パーティション (Partition) の秘密鍵が格納されているブロックへのポインタ

\* PRI\_PAR Size : パーティション (Partition) の秘密鍵のサイズ

\* PARAM\_PAR Pointer : パーティション (Partition) の公開鍵パラメータが格納されているブロックへのポインタ

\* PARAM\_PAR Size : パーティション (Partition) の公開鍵パラメータのサイズ

\* CERT\_PAR Pointer : 認証局 CA (PAR) が発行したパーティション (Partition) の公開鍵証明書が格納されているブロックへのポインタ

\* CERT\_PAR Size : 認証局 CA (PAR) が発行したパーティション (Partition) の公開鍵証明書のサイズ

\* CRL\_PAR Pointer : パーティション (Partition) のリボケーションリスト (Revocation List) が格納されて

いるブロックへのポインタ

\* CRL\_PAR Size : パーティション (Partition) のリボケーションリスト (Revocation List) のサイズ

\* FRTIC (FRT Issuer Category) : ファイル登録チケット (FRT) 発行者カテゴリ

\* FRTIC Version : ファイル登録チケット (FRT) 発行者カテゴリ (FRTIC) のバージョン

\* DUTIC\_PAR (DUT Issuer Category) : データアップデートチケット (DUT) 発行者カテゴリ

\* DUTIC\_PAR Version : データアップデートチケット (DUT) 発行者カテゴリ (DUTIC) のバージョン

【0184】図22は、共通鍵系パーティション鍵情報ブロック (Partition Key Definition Block(Common)) データ構成を示す。共通鍵系パーティション鍵情報ブロック (Partition Key Definition Block(Common)) には以下のデータが格納される。

【0185】\* Mkauth\_PAR\_A Pointer : 双方向個別鍵認証用マスター鍵 (Mkauth\_PAR\_A) のポインタ

\* Mkauth\_PAR\_A Size : 双方向個別鍵認証用マスター鍵 (Mkauth\_PAR\_A) のサイズ

\* Kauth\_PAR\_B Pointer : 双方向個別鍵認証用鍵 (Kauth\_PAR\_B) のポインタ

\* Kauth\_PAR\_B Size : 双方向個別鍵認証用鍵 (Kauth\_PAR\_B) のサイズ

\* Kfrt Pointer : ファイル登録チケット (FRT) のMAC検証用鍵 (Kfrt) が格納されているブロックへのポインタ

\* Kfrt Size : ファイル登録チケット (FRT) のMAC検証用鍵 (Kfrt) のサイズ

\* Kdut\_PAR1-4 Pointer : データアップデートチケット (DUT) のMAC検証用鍵 (Kdut) が格納されているブロックへのポインタ

\* Kdut\_PAR1-4 Size : データアップデートチケット (DUT) のMAC検証用鍵 (Kdut) のサイズ

\* IRL\_PAR Pointer : パーティション (Partition) の排除デバイスのIDを格納したリボケーションリスト (Revocation List-Device ID) が格納されているブロックへのポインタ

\* IRL\_PAR Size : パーティション (Partition) のリボケーションリスト (Revocation List-Device ID) のサイズ

【0186】上述のデータ中に示される双方向個別鍵認証の方法、MAC (Message Authenticate Code) 検証処理については、後段で詳細に説明する。また、Kdut\_PAR は4種類存在し、(Kdut\_PAR1, Kdut\_PAR2), (Kdut\_PAR3, Kdut\_PAR4) のペアで使われる。例えば、Kdut\_PAR1, 3 はMAC生成用、Kdut\_PAR2, 4は暗号用に使われる。

【0187】図23は、パーティション鍵領域 (Partition Key Area) のデータ構成を示す。パーティション鍵領域 (Partition Key Area) には以下のデータが格納さ

れる。なお、パーティション鍵領域 (Partition Key Area) の各格納鍵には、バージョン情報が併せて格納される。鍵の更新時には、バージョンについても併せて更新される。

【0188】\* IRL\_PAR : パーティションアクセス排除デバイス (Device) 、排除機器 (デバイスアクセス機器としてのリーダライタ、PC等のチケットユーザ、チケット発行手段) の識別子 (ID) を登録したリボケーションリスト (Revocation List (Device ID))

\* CRL\_PAR : パーティションアクセス排除デバイス (Device) 、排除機器 (デバイスアクセス機器としてのリーダライタ、PC等のチケットユーザ、チケット発行手段) の公開鍵証明書識別子 (ex. シリアルナンバ: SN) を登録したリボケーションリスト (Revocation List (Certificate))

\* Kdut\_PAR1 : データアップデートチケット (DUT) のMAC検証用鍵

\* Kdut\_PAR2 : データ更新用暗号鍵

\* Kdut\_PAR3 : データアップデートチケット (DUT) のMAC検証用鍵

\* Kdut\_PAR4 : データ更新用暗号鍵

\* Kfrt : ファイル登録チケット (FRT) のMAC検証用鍵

\* CERT\_PAR : 認証局CA (PAR) が発行したパーティション (Partition) の公開鍵証明書

\* PRI\_PAR : パーティション (Partition) の秘密鍵

\* PARAM\_PAR : パーティション (Partition) の公開鍵パラメータ

\* PUB\_CA (PAR) : 認証局CA (PAR) の公開鍵

\* Mkauth\_PAR\_A : 双方向個別鍵認証用マスター鍵

\* Kauth\_PAR\_B : 双方向個別鍵認証用共通鍵

【0189】図24は、ファイル定義ブロック (FDB : File Definition Block) のデータ構成を示す。ファイル定義ブロック (File Definition Block) には以下のデータが格納される。

【0190】\* File ID : ファイル (File) 識別名

\* File Start Position : ファイル (File) スタートアドレス

\* File Size : ファイル (File) サイズ

\* SPTIC (SPT Issuer Category) : サービス許可チケット (SPT) 発行者カテゴリ

\* SPTIC Version : サービス許可チケット (SPT) 発行者カテゴリ (SPTIC) のバージョン

\* File Structure Type Code : ファイル構造タイプ (File Structure Type) のコード

\* Acceptable Authentication Type : 許容認証タイプを示す。各ファイル構造タイプ (File Structure Type) に対して定義されるアクセスモードとこのフィールドの各ビット (本例では最大16個) が対応する。詳細は下記に説明する。

\* Acceptable Verification Type:許容検証タイプを示す。各ファイル構造タイプ (File Structure Type) に対して、定義されるアクセスモードとこのフィールドの各ビット (本例では最大16個) が対応する。詳細は下記に説明する。

\* Kspt :サービス許可チケット (SPT) のMAC検証用鍵(Kspt)

【0191】上記の許容認証タイプ (Acceptable Authentication Type) は、各ファイル構造タイプ (File Structure Type) に対して定義されるアクセスモードとこのフィールドの各ビット (本例では最大16個) が対応するように設定された許容認証タイプであり、例えばあるアクセスモードを実行する際に、そのモードに対応するビットに1がたっている場合には、公開鍵認証が済んで認証済みでないとい実行されないものとする。これにより、より重要度の高いコマンド (例えば入金処理など) の実行の際には、公開鍵認証を義務づけ、安全性を確保できる。チケットを用いることで同様の制御も可能ではあるが、許容認証タイプ (Acceptable Authentication Type) は、チケットと異なり、ファイル定義ブロック (FDB: File Definition Block) の一部としてデバイスに格納されることになるため、この情報はファイル生成後に変更されることがない。従って、絶対に許容認証タイプの変更を許さない強い制約を与えたいときに利用することにより、安全性の最低限の保証を与えることができる。

【0192】また上記の許容検証タイプ (Acceptable Verification Type) は、各ファイル構造タイプ (File Structure Type) に対して定義されるアクセスモードとこのフィールドの各ビット (本例では最大16個) が対応するように設定された許容検証タイプであり、例えばあるアクセスモードを実行する際に、そのモードに対応するビットに1がたっている場合には、公開鍵方式によるチケット検証が済んでないとい実行されないものとする。この例では、各フィールドを2バイトづつにしたため、最大16個のアクセスモードとの対応付けしかできないが、必要に応じてフィールドサイズを大きくとることにより、より多くのコマンドに対応付ける構成とすることができる。

【0193】また、本実施例構成においては、許容認証タイプ (Acceptable Authentication Type) 、許容検証タイプ (Acceptable Verification Type) は設定が「1」のときに公開鍵方式の認証または検証を必要とする設定としてあるが、これらの各フィールドを2ビット単位の構成として、値が「11」の場合には公開鍵方式、「01」の場合には共通鍵方式、「00」「10」の場合には公開鍵方式、共通鍵方式のいずれでもは許容する、などの細分化した設定としてもよい。

【0194】上述のデータ中のファイル構造タイプ (File Structure Type) は、パーティション内に生成され

るファイルの構造を示すコードである。ファイル構造とコードの対応の一例を図25に示す。

【0195】ファイル構造には、図25に示す各種構造 (File Structure) があり、それぞれにコード0001~0007が割り当てられる。各構造の意味を以下に示す。

【0196】\* Random :本ファイル構造を持つデータはすべての読書きがランダムに可能なファイルである。

\* Purse :本ファイル構造を持つデータは、金額情報データであり、減算 (Sub) 、加算 (Add) など金額の価値変更処理が可能であるデータファイルである。

\* Cyclic :本ファイル構造を持つデータは循環型 (Cyclic) のデータ書き込みが可能なファイル構造である。

\* Log :本ファイル構造を持つデータは、ログデータファイルであり、各データ処理情報についての記録情報ファイルである。

\* Key :本ファイル構造を持つデータファイルは、鍵情報ファイルであることを示す。

\* 複合ファイル :上記各種ファイル構造の複合構造 (Ex. PurseとLog) を持つファイルである。複合ファイルには、その組み合わせパターンにより異なるコード (図では0006:複合ファイル1、0007:複合ファイル2) が割り当てられる。

【0197】以上、デバイスのメモリ部に格納されるデータについて説明した。これらのデータを用いた具体的な処理については、後段で説明する。

【0198】[A8. 各チケットのデータフォーマット] 前述したように、デバイスに対するパーティションの設定登録処理には、正当なチケット発行手段 (Ticket Issuer) の発行したパーティション登録チケット (PRT: Partition Registration Ticket) 、デバイスに設定されたパーティション内に対するファイルの設定登録処理には、正当なチケット発行手段 (Ticket Issuer) の発行したファイル登録チケット (FRT: File Registration Ticket) 、また、各ファイルに対するアクセスには正当なチケット発行手段 (Ticket Issuer) の発行したサービス許可チケット (SPT: Service Permission Ticket) が必要となる。また、前述のデバイスのメモリ部のデータ説明の欄で簡単に説明したように、デバイス格納データの更新処理にはデータアップデートチケット (DUT) を必要とする。

【0199】これらの各チケットはデバイスに対するアクセスルールをバイナリデータとして記述したデータ列によって構成される。チケットはデバイスに対する処理に応じて、チケットユーザである例えばデバイスアクセス機器としてのリーダーライターからデバイスに送信される。チケットを受信したデバイスはチケットの正当性検証処理を実行し、正当性検証に成功した場合、チケットに記録されたルールに従って各種の処理 (ex. パーティション生成、ファイル生成、データアクセス) が実行

される。以下、これらの各チケットのデータフォーマットについて説明する。

【0200】(A8. 1. パーティション登録チケット (PRT)) パーティション登録チケット (PRT: Partition Registration Ticket) は、デバイスに対するパーティションの設定登録処理の際に適用されるアクセスコントロールチケットである。正当なデバイスマネージャ管轄下のチケット発行手段 (Ticket Issuer) の発行したPRTを用い、PRTに記録された手続きに従って、パーティションマネージャの管轄下のチケットユーザ (ex. デバイスアクセス機器としてのリーダライタ) によりデバイスにアクセスすることで、PRTに記録された制限内でパーティションを設定することができる。

【0201】図26にパーティション登録チケット (PRT: Partition Registration Ticket) のデータフォーマットを示す。パーティション登録チケット (PRT: Partition Registration Ticket) には以下に説明するデータが格納される。

【0202】\* Ticket Type : チケット (Ticket) の種別。

\* Format Version : チケット (Ticket) のフォーマットバージョン

\* Ticket Issuer : デバイスマネージャの識別子 (=DMC)

\* Serial Number : チケット (Ticket) のシリアル番号

\* Size of Ticket : チケット (Ticket) のサイズ

\* Authentication Flag : チケット (Ticket) の利用処理においてデバイス (Device) との相互認証が必要か否かを示すフラグ

\* Ticket User の所属(Group) : チケット (Ticket) 利用者の所属

\* Authentication Type : デバイス (Device) の相互認証のタイプ (公開鍵認証、または、共通鍵認証、または、いずれでも可 (Any))

\* Ticket User の識別子 : チケット (Ticket) 利用者を判別する識別データ (カテゴリまたは識別子)

当フィールドは、[Authentication Type] と連携したデータとされ、[Authentication Type] が公開鍵認証の場合: 識別名 (DN: Distinguished Name) またはカテゴリ (Category) またはシリアル番号 (SN) が格納され、共通鍵認証の場合、: 認証IDが格納される。認証不要の場合は格納は必須ではない。

\* PMC: パーティションマネージャコード (Partition Manager Code) として、パーティション定義ブロック (Partition Definition Block) に記述されるコード

\* PMC Version : パーティションマネージャコード (PMC) のバージョン

\* Operation Type : パーティション (Partition) 作成か削除かの指定 (作成 (Generate) / 削除 (Delete))

\* Partition Size : パーティション (Partition) のサイズ

\* Integrity Check Type : チケット (Ticket) の正当性検証値の種別 (公開鍵方式 (Public) / 共通鍵方式 (Common))

\* Integrity Check Value : チケット (Ticket) の正当性検証値 (公開鍵方式: 署名 (Signature)、共通鍵方式: MAC)

【0203】なお、パーティション登録チケット (PRT) 発行手段 (PRT Issuer) の発行したチケット (Ticket) を、チケットユーザに対して送信する際には、公開鍵方式の場合、パーティション登録チケット (PRT) 発行手段 (PRT Issuer) の公開鍵証明書 (CERT\_PRTI) も一緒に送信する。PRT発行手段の公開鍵証明書 (CERT\_PRTI) の属性 (Attribute) は、PRT発行手段 (PRT Issuer) の識別子 (PRTIC) と一致する。

【0204】デバイス (Device) の相互認証のタイプ (公開鍵認証、または、共通鍵認証、または、いずれでも可 (Any)) を記録した [Authentication Type] には、チケットを使用した相互認証として実行すべき認証タイプが記録される。具体的には、後段で詳細に説明するが、デバイス認証、パーティション認証のいずれか、または両方の認証を実行する指定、また公開鍵方式、共通鍵方式のどちらを実行するか、またはいずれの認証でも可能であるかについての情報が記録される。

【0205】チケット (Ticket) の正当性検証値 (公開鍵方式: 署名 (Signature)、共通鍵方式: MAC) を記録する [Integrity Check Value] フィールドには、公開鍵方式であれば、パーティション登録チケット発行手段 (PRT Issuer) の秘密鍵に基づく署名 (図12参照) が生成され格納される。デバイスマネージャ自体がパーティション登録チケット発行手段 (PRT Issuer) を兼ねる場合は、デバイスマネージャの秘密鍵を用いて署名が生成される。署名検証処理 (図13参照) の際は、対応のCA (DEV) の公開鍵が用いられる。従って、チケット検証を実行するデバイスは、チケット受領に際し、または前もってパーティション登録チケット発行手段 (PRT Issuer) (ex. デバイスマネージャ) の公開鍵 (公開鍵証明書) を取得することが必要である。

【0206】パーティション登録チケット (PRT) 発行手段 (PRT Issuer) の公開鍵証明書 (CERT\_PRTI) の検証の後、公開鍵証明書 (CERT\_PRTI) から取り出したパーティション登録チケット (PRT) 発行手段 (PRT Issuer) の公開鍵によりICV (Integrity Check Value) の署名検証が可能となる。これらの処理については、フローを用いて後段で説明する。

【0207】(A8. 2. ファイル登録チケット (FRT)) ファイル登録チケット (FRT: File Registration Ticket) は、デバイスに対して設定されたパーティ

ションにファイルを登録する際に適用されるアクセスコントロールチケットである。正当なパーティションマネージャ管轄下のチケット発行手段 (Ticket Issuer) の発行したFRTを用い、FRTに記録された手続きに従ってチケットユーザ (ex. デバイスアクセス機器としてのリーダライタ) によりデバイスにアクセスすることで、FRTに記録された制限内でファイルを設定することができる。

【0208】図27にファイル登録チケット (FRT : File Registration Ticket) のデータフォーマットを示す。ファイル登録チケット (FRT : File Registration Ticket) には以下に説明するデータが格納される。

【0209】\* Ticket Type : チケット (Ticket) の種別

\* Format Version : チケット (Ticket) のフォーマットバージョン

\* Ticket Issuer : パーティションマネージャの識別子 (= PMC)

\* Serial Number : チケット (Ticket) のシリアル番号

\* Size of Ticket : チケット (Ticket) のサイズ

\* Authentication Flag : チケット (Ticket) の利用処理においてデバイス (Device) との相互認証が必要か否かを示すフラグ

\* Ticket User の所属 (Group) : チケット (Ticket) 利用者の所属

\* Authentication Type : デバイス (Device) の相互認証のタイプ (公開鍵認証、または、共通鍵認証、または、いずれでも可 (Any))

\* Ticket User の識別子 : チケット (Ticket) 利用者を判別する識別データ (カテゴリまたは識別子)

当フィールドは、[Authentication Type] と連携したデータとされ、[Authentication Type] が公開鍵認証の場合 : 識別名 (DN : Distinguished Name) またはカテゴリ (Category) またはシリアル番号 (CN) が格納され、共通鍵認証の場合、: 認証IDが格納される。認証不要の場合は格納は必須ではない。

\* SPTIC : サービス許可チケット発行手段のコード

\* SPTIC Ver : サービス許可チケット発行手段のコード (SPTIC) のバージョン

\* File ID : パーティション内に生成するファイル (File) の識別子 (ID)

\* Operation Type : ファイルの作成か削除かの指定 (生成 (Generate) / 削除 (Delete))

\* File Size : 生成するファイル (File) のサイズ

\* File Structure : 生成するファイル (File) のファイル構造 (Structure)

\* Acceptable Authentication Type : このチケットで定義されるファイルに対するアクセスモードを実行するために必要とする相互認証の種類 (公開鍵方式、公開鍵、共通鍵いずれでも可) を表すビット列

\* Acceptable Verification Type : このチケットで定義されるファイルに対するアクセスモードを実行するために必要とするサービス許可チケット (SPT) の検証の種類 (公開鍵方式、公開鍵、共通鍵いずれでも可) を表すビット列

\* Kspt\_Encrypted : ファイル定義ブロック (File Definition Block) に記載されるサービス許可チケット (SPT) のMAC検証用鍵 Kspt をそのパーティションのファイル登録チケットのMAC検証用鍵 Kfrt で暗号化したデータ Kfrt (Kspt)

\* Integrity Check Type : チケット (Ticket) の正当性検証値の種別 (公開鍵方式 (Public) / 共通鍵方式 (Common))

\* Integrity Check Value : チケット (Ticket) の正当性検証値 (公開鍵方式 : 署名 (Signature)、共通鍵方式 : MAC)

【0210】なお、ファイル登録チケット (FRT) 発行手段 (FRT Issuer) の発行したチケット (Ticket) を、チケットユーザに対して送信する際には、公開鍵方式の場合、ファイル登録チケット (FRT) 発行手段 (FRT Issuer) の公開鍵証明書 (CERT\_FRTI) も一緒に送信する。FRT発行手段の公開鍵証明書 (CERT\_FRTI) の属性 (Attribute) は、ファイル登録チケット (FRT) 発行手段 (FRT Issuer) の識別子 (FRTIC) と一致する。

【0211】デバイス (Device) の相互認証のタイプ (公開鍵認証、または、共通鍵認証、または、いずれでも可 (Any)) を記録した [Authentication Type] には、チケットを使用した相互認証として実行すべき認証タイプが記録される。具体的には、後段で詳細に説明するが、デバイス認証、パーティション認証のいずれか、または両方の認証を実行する指定、また公開鍵方式、共通鍵方式のどちらを実行するか、またはいずれの認証でも可能であるかについての情報が記録される。

【0212】チケット (Ticket) の正当性検証値 (公開鍵方式 : 署名 (Signature)、共通鍵方式 : MAC) を記録する [Integrity Check Value] フィールドには、公開鍵方式であれば、ファイル登録チケット発行手段 (FRT Issuer) の秘密鍵に基づく署名 (図12参照) が生成され格納される。パーティションマネージャ自体がファイル登録チケット発行手段 (FRT issuer) を兼ねる場合は、パーティションマネージャの秘密鍵を用いて署名が生成される。署名検証処理 (図13参照) の際は、ファイル登録チケット発行手段の公開鍵が用いられる。従って、チケット検証を実行するデバイスは、チケット受領に際し、または前もってファイル登録チケット発行手段 (FRT Issuer) (ex. パーティションマネージャ) の公開鍵 (公開鍵証明書) を取得することが必要である。

【0213】ファイル登録チケット (FRT) 発行手段

(FRT Issuer) の公開鍵証明書 (CERT\_FRTI) の検証の後、公開鍵証明書 (CERT\_FRTI) から取り出したファイル登録チケット (FRT) 発行手段 (FRT Issuer) の公開鍵により ICV (IntegrityCheck Value) の署名検証が可能となる。これらの処理については、フローを用いて後段で説明する。

【0214】(A8. 3. サービス許可チケット (SPT)) サービス許可チケット (SPT: Service Permission Ticket) は、デバイスに対して設定されたパーティション内の各データに対してアクセスしてデータ読み出し、データ書き込み、金額データの減算、加算などの処理を実行する際に適用されるアクセスコントロールチケットである。正当なパーティションマネージャ管轄下のチケット発行手段 (Ticket Issuer) の発行した SPT を使い、SPT に記録された手続きに従ってチケットユーザ (ex. デバイスアクセス機器としてのリーダライタ) によりデバイスにアクセスすることで、SPT に記録された制限内でデータ処理を実行することができる。

【0215】なお、サービス許可チケット (SPT: Service Permission Ticket) は、パーティションに設定されたファイルの中から唯一のファイルに対してのみアクセスを許可する形式と、複数のファイルに対するアクセスを許可する形式とがあり、それぞれの形式について説明する。

【0216】図28に、パーティションに設定されたファイルの中から唯一のファイルに対してのみアクセスを許可する形式のサービス許可チケット (SPT: Service Permission Ticket) のデータフォーマットを示す。サービス許可チケット (SPT: Service Permission Ticket) には以下に説明するデータが格納される。

【0217】\* Ticket Type : チケット (Ticket) の種別。

\* Format Version : チケット (Ticket) のフォーマットバージョン

\* Ticket Issuer : パーティションマネージャの識別子 (=PMC)

\* Serial Number : チケット (Ticket) のシリアル番号

\* Size of Ticket : チケット (Ticket) のサイズ

\* Authentication Flag : チケット (Ticket) の利用処理においてデバイス (Device) との相互認証が必要か否かを示すフラグ

\* Ticket User の所属(Group) : チケット (Ticket) 利用者の所属

\* Authentication Type : デバイス (Device) の相互認証のタイプ (公開鍵認証、または、共通鍵認証、または、いずれでも可 (Any))

\* Ticket User の識別子 : チケット (Ticket) 利用者を判別する識別データ (カテゴリまたは識別子)

当フィールドは、[Authentication Type] と連携した

データとされ、[Authentication Type] が公開鍵認証の場合：識別名 (DN: Distinguished Name) またはカテゴリ (Category) またはシリアル番号 (CN) が格納され、共通鍵認証の場合、: 認証IDが格納される。認証不要の場合は格納は必須ではない。

\* File ID : パーティション内のアクセスファイル (File) の識別子 (ID)

\* File Access Mode : アクセスを許諾するファイル (File) へのアクセスモード (Access Mode)

\* Integrity Check Type : チケット (Ticket) の正当性検証値の種別 (公開鍵方式 (Public) / 共通鍵方式 (Common))

\* Integrity Check Value : チケット (Ticket) の正当性検証値 (公開鍵方式: 署名 (Signature)、共通鍵方式: MAC)

【0218】なお、サービス許可チケット (SPT) 発行手段 (SPT Issuer) の発行したチケット (Ticket) を、チケットユーザに対して送信する際には、公開鍵方式の場合、サービス許可チケット (SPT) 発行手段 (SPT Issuer) の公開鍵証明書 (CERT\_SPTI) も一緒に送信する。SPT 発行手段の公開鍵証明書 (CERT\_SPTI) の属性 (Attribute) は、(SPT) 発行手段 (SPT Issuer) の識別子 (SPTIC) と一致する。

【0219】サービス許可チケット (SPT) 発行手段 (SPT Issuer) を、パーティションマネージャが兼ねる構成においては、サービス許可チケット (SPT) 発行手段 (SPT Issuer) のコードは、パーティションマネージャコード (PMC) として設定することが可能である。

【0220】デバイス (Device) の相互認証のタイプ (公開鍵認証、または、共通鍵認証、または、いずれでも可 (Any)) を記録した [Authentication Type] には、チケットを使用した相互認証として実行すべき認証タイプが記録される。具体的には、後段で詳細に説明するが、デバイス認証、パーティション認証のいずれか、または両方の認証を実行する指定、また公開鍵方式、共通鍵方式のどちらを実行するか、またはいずれの認証でも可能であるかについての情報が記録される。

【0221】チケット (Ticket) の正当性検証値 (公開鍵方式: 署名 (Signature)、共通鍵方式: MAC) を記録する [Integrity Check Value] フィールドには、公開鍵方式であれば、サービス許可チケット発行手段 (SPT Issuer) の秘密鍵に基づく署名 (図12参照) が生成され格納される。パーティションマネージャ自体がサービス許可チケット発行手段 (SPT Issuer) を兼ねる場合は、パーティションマネージャの秘密鍵を用いて署名が生成される。署名検証処理 (図13参照) の際は、サービス許可チケット (SPT) 発行手段 (SPT Issuer) の公開鍵が用いられる。従って、チケット検証を実行するデバイスは、チケット受領に際し、ま



たは前もってサービス許可チケット発行手段 (SPT Issuer) (e x. パーティションマネージャ) の公開鍵 (公開鍵証明書) を取得することが必要である。

【0222】サービス許可チケット (SPT) 発行手段 (SPT Issuer) の公開鍵証明書 (CERT\_SPTI) の検証の後、公開鍵証明書 (CERT\_SPTI) から取り出したサービス許可チケット (SPT) 発行手段 (SPT Issuer) の公開鍵により I C V (IntegrityCheck Value) の署名検証が可能となる。これらの処理については、フローを用いて後段で説明する。

【0223】上述のチケット・フォーマットの説明中、File Access Mode :アクセスを許諾するファイル (File) へのアクセスモード (Access Mode) に記録されるモードとアクセス態様について、図29を用いて説明する。

【0224】ファイルとして生成されるデータは、ユーザの識別データ、金額データ、暗号処理鍵データ、ログデータ、あるいは複合ファイルデータなど様々であり、各データに応じたアクセス処理、すなわちデータ読み取り、書き込み、消去、加算、減算、暗号化、復号…がアクセスデータに対して実行されることになる。

【0225】サービス許可チケット (SPT) のFile Access Modelには、このような様々なアクセスの態様中、いずれのアクセスモードを許可しているものかを定義している。アクセスモードの一覧を図29に示す。図29に示すアクセスモードは一例であり、この他にもデバイスに格納されるデータに応じたアクセスモードを設定することができる。

【0226】サービス許可チケット (SPT) に設定された [File ID :パーティション内のアクセスファイル (File) の識別子 (ID)] によって示されるファイルに対してファイルアクセスモード [File Access Mode] に定義された処理が実行できる。サービス許可チケット (SPT) に設定されたファイルアクセスモードが読み出し (Read) であればファイル内データの読み出しが実行できる。サービス許可チケット (SPT) に設定されたファイルアクセスモードが書き込み (Write) であればファイル内へのデータの書き込みが実行できる。

【0227】このようなアクセスモードは、前述したファイル構造 (File Structure) (図25参照) によって設定可能なモードが制限される。例えばファイル構造が purseであれば金額データであり、加算 (add)、減算 (Sub) 等のアクセスモードの設定が可能となる。このようなファイル構造と、設定可能なアクセスモード、さらに、デバイスアクセス機器としてのリーダライタからデバイスに対して送信されるコマンドの例を図30に示す。

【0228】図30には、ファイル構造が Randomの場合と、複合ファイルの場合に設定可能なアクセスモード、およびコマンド例を示している。

【0229】例えばファイル構造が Randomの場合において、アクセスモードが読み出し (Read) である場合、デバイスが許容可能なコマンドは [Read] のみとなる。また、同様にファイル構造が Randomの場合において、アクセスモードが暗号化読み出し (Read) である場合、デバイスが許容可能なコマンドは暗号化読み出し [EncRead] のみとなる。

【0230】また、ファイル構造がPurseおよびLogを含むような複合ファイルの場合において、アクセスモードが入金系である場合、デバイスが許容可能なコマンドは預け入れ [Deposit] のみとなる。また、同様にファイル構造がPurseおよびLogを含むような複合ファイルの場合において、アクセスモードが出金系である場合、デバイスが許容可能なコマンドは引き出し [Withdraw]、レシート生成 [Make Receipt]、レシート読み出し [Read Receipt] となる。

【0231】ファイルアクセスモード (図29参照) の入金系に対応する許容コマンド (図30参照) として、上述の預け入れコマンド (Deposit Command) を定義し、アクセス許可チケットのファイルアクセスモード (File Access Mode) に [入金系] を設定し、ファイル ID (File ID) として、電子マネーを構成する複合ファイルを指定したアクセス許可チケット (SPT) を生成して、ファイルアクセス装置からデバイスに対して送信し、預け入れコマンド (Deposit Command) とともに、預け入れ金額データを送信することにより、例えば、複合ファイル中のファイル [Purse] にX円を加算し、複合ファイル中のファイル [Log] に処理記録を書き込むなどのシーケンシャル処理を実行させることが可能となる。これらの処理についての詳細は、後述する。

【0232】図30に示す他にも、様々なアクセスモード、コマンドの組み合わせが可能であり、実際のデバイスの利用形態に応じて設定されることになる。

【0233】デバイスは、メモリ部に格納された各ファイルに対して許容されるコマンドの定義データを図30のようなテーブルとして保有し、前記アクセス機器から入力されるコマンドが前記定義データに定義されたコマンドである場合にのみコマンドを実行する。複合ファイルに対して許容されるコマンドの定義データには、上述したように複合ファイルに含まれる複数ファイルの各々に対応して実行可能な複数のコマンドからなるシーケンスコマンドを含む。

【0234】処理対象となる特定のファイルをサービス許可チケット (SPT) のファイルID (File ID) 欄に設定し、所定のアクセスモードをサービス許可チケット (SPT) のファイルアクセスモード (File Access Mode) に設定することで、当該サービス許可チケット (SPT) を利用したファイルアクセスをコントロールする

ことが可能となる。具体的処理については、後段でフローを用いて説明する。

【0235】次に、図31に、パーティションに設定されたファイル中の複数ファイルに対してアクセスを許可する形式のサービス許可チケット (SPT: Service Permission Ticket) のデータフォーマットを示す。サービス許可チケット (SPT: Service Permission Ticket) には以下に説明するデータが格納される。

【0236】\* Ticket Type : チケット (Ticket) の種別。

\* Format Version : チケット (Ticket) のフォーマットバージョン

\* Ticket Issuer : パーティションマネージャの識別子 (= PMC)

\* Serial Number : チケット (Ticket) のシリアル番号

\* Size of Ticket : チケット (Ticket) のサイズ

\* Authentication Flag : チケット (Ticket) の利用処理においてデバイス (Device) との相互認証が必要か否かを示すフラグ

\* Ticket User の所属 (Group) : チケット (Ticket) 利用者の所属

\* Authentication Type : デバイス (Device) の相互認証のタイプ (公開鍵認証、または、共通鍵認証、または、いずれでも可 (Any))

\* Ticket User の識別子 : チケット (Ticket) 利用者を判別する識別データ (カテゴリまたは識別子)

当フィールドは、[Authentication Type] と連携したデータとされ、[Authentication Type] が公開鍵認証の場合: 識別名 (DN: Distinguished Name) またはカテゴリ (Category) が格納され、共通鍵認証の場合、: 認証IDが格納される。認証不要の場合は格納は必須ではない。

\* File ID : パーティション内のアクセスファイル (File) の識別子 (ID)

\* File Access Mode : アクセスを許諾するファイル (File) へのアクセスモード (Access Mode)

\* Group of Target File : アクセスを許すファイル (File) のグループ (Group)

\* Target File ID : アクセスを許すファイル (File) の識別子 (ID)

\* Read/Write Permission : アクセスを許すファイル (File) (ターゲットファイル (Target File)) に対する処理態様 (読み出し (Read)、書き込み (Write)) の許可

\* Integrity Check Type : チケット (Ticket) の正当性検証値の種別 (公開鍵方式 (Public) / 共通鍵方式 (Common))

\* Integrity Check Value : チケット (Ticket) の正当性検証値 (公開鍵方式: 署名 (Signature)、共通鍵方式: MAC)

【0237】このように、Group of Target File : アクセスを許すファイル (File) のグループ (Group) を定義してチケットに記録することにより、パーティション内の複数のファイルに対するアクセスを唯一のサービス許可チケット (SPT) で許可することが可能となる。

【0238】なお、上述のサービス許可チケット (SPT) 発行手段 (SPT Issuer) の発行したチケット (Ticket) を、チケットユーザに対して送信する際にも、公開鍵方式の場合、サービス許可チケット (SPT) 発行手段 (SPT Issuer) の公開鍵証明書 (CERT\_SPTI) も一緒に送信する。SPT発行手段の公開鍵証明書 (CERT\_SPTI) の属性 (Attribute) は、サービス許可チケット (SPT) 発行手段 (SPT Issuer) の識別子 (SPTIC) と一致する。

【0239】デバイス (Device) の相互認証のタイプ (公開鍵認証、または、共通鍵認証、または、いずれでも可 (Any)) を記録した [Authentication Type] には、チケットを使用した相互認証として実行すべき認証タイプが記録される。具体的には、後段で詳細に説明するが、デバイス認証、パーティション認証のいずれか、または両方の認証を実行する指定、また公開鍵方式、共通鍵方式のどちらを実行するか、またはいずれの認証でも可能であるかについての情報が記録される。

【0240】サービス許可チケット (SPT) 発行手段 (SPT Issuer) の公開鍵証明書 (CERT\_SPTI) の検証の後、公開鍵証明書 (CERT\_SPTI) から取り出したサービス許可チケット (SPT) 発行手段 (SPT Issuer) の公開鍵によりICV (IntegrityCheck Value) の署名検証が可能となる。これらの処理については、フローを用いて後段で説明する。

【0241】(A8. 4. データアップデートチケット (DUT)) データアップデートチケット (DUT: Data Update Ticket) は、デバイスに格納された様々なデータに対してアクセスしてデータの更新処理を実行する際に適用されるアクセスコントロールチケットである。正当なデータアップデートチケット (DUT) 発行手段 (Ticket Issuer) の発行したDUTを用い、DUTに記録された手続きに従ってチケットユーザ (ex. デバイスアクセス機器としてのリーダライタ) によりデバイスにアクセスすることで、DUTに記録された制限内でデータ処理を実行することができる。

【0242】なお、データアップデートチケット (DUT: Data Update Ticket) は、デバイスマネージャの管理するデータ項目の更新処理を実行するために適用されるチケットDUT (DEV) と、パーティションマネージャの管理するパーティション内のデータ項目の更新処理を実行するために適用されるチケットDUT (PAR) がある。チケット: DUT (DEV) 発行手段はデバイスマネージャの管理下にあり、チケット: DUT (PAR) 発行手段はパーティションマネージャの管理

下にある。

【0243】図32に、2つのデータアップデートチケット (DUT : Data Update Ticket)、DUT (DEV)、DUT (PAR) のデータフォーマットを示す。データアップデートチケット (DUT : Data Update Ticket) には以下に説明するデータが格納される。

【0244】\* Ticket Type : チケット (Ticket) の種別 (DUT (DEV) / DUT (PAR))

\* Format Version : チケット (Ticket) のフォーマットバージョン

\* Ticket Issuer : デバイス / パーティションマネージャの識別子。チケット (Ticket) の種別 (Ticket Type) が DUT (DEV) なら DMC、DUT (PAR) なら PMC となる

\* Serial Number : チケット (Ticket) のシリアル番号

\* Size of Ticket : チケット (Ticket) のサイズ

\* Ticket User の所属 (Group) : チケット (Ticket) 利用者の所属

\* Ticket User の識別子 : チケット (Ticket) 利用者を判別する識別データ (カテゴリまたは識別子)

当フィールドは、[Authentication Type] と連携したデータとされ、[Authentication Type] が公開鍵認証の場合 : 識別名 (DN : Distinguished Name) またはカテゴリ (Category) が格納され、共通鍵認証の場合、: 認証 ID が格納される。認証不要の場合は格納は必須ではない。

\* Authentication Type : デバイス (Device) の相互認証のタイプ (公開鍵認証、または、共通鍵認証、または、いずれでも可 (Any))

\* Encrypted Flag : 更新されるデータが暗号化されているか否か (暗号化 : Encrypted / 非暗号化 : none)

\* Old Data Code : 更新される古いデータのコード (Code)

\* Data Version Rule : データ更新をする時のバージョン条件

\* Data Version Condition : データ更新をする時のバージョン値

\* Size of New Data : 更新する新しいデータのサイズ

\* New Data : 更新する新しいデータ (暗号化される場合もある)。

\* New Data Version : 更新するデータのバージョン

\* Integrity Check Type : チケット (Ticket) の正当性検証値の種別 (公開鍵方式 (Public) / 共通鍵方式 (Common))

\* Integrity Check Value : チケット (Ticket) の正当性検証値 (公開鍵方式 : 署名 (Signature)、共通鍵方式 : MAC)

【0245】データアップデートチケット (DUT : Data Update Ticket) を適用したデータ更新をする際に、[Data Version Rule : データ更新をする時のバージョン

条件] と、[Data Version Condition : データ更新をする時のバージョン値]、これら2つのフィールドの組み合わせにより条件を表現する。

【0246】データ更新をする時のバージョン条件 [Data Version Rule] は、Any, Exact, Older の3種類が存在する。Any はバージョン (Version) 条件に無関係でデータ更新が可能、Exact は、続く [Data Version Condition] に指定された値と同じ場合にデータ更新が可能、Older は、New Data Version の方が新しい場合にのみデータ更新が可能となる。なお、バージョン条件 [Data Version Rule] が Any, または Older の場合は、[Data Version Condition] は使用しないかもしくは無視する。

【0247】デバイス (Device) の相互認証のタイプ (公開鍵認証、または、共通鍵認証、または、いずれでも可 (Any)) を記録した [Authentication Type] には、チケットを使用した相互認証として実行すべき認証タイプが記録される。具体的には、後段で詳細に説明するが、デバイス認証、パーティション認証のいずれか、または両方の認証を実行する指定、また公開鍵方式、共通鍵方式のどちらを実行するか、またはいずれの認証でも可能であるかについての情報が記録される。

【0248】データアップデートチケット-DUT (DEV) 発行手段 (DUT Issuer) を、デバイスマネージャが兼ねる構成においては、データアップデートチケット-DUT (DEV) 発行手段 (DUT Issuer) のコード (チケットユーザ (Ticket User)) は、デバイスマネージャコード (DMC) として設定することが可能である。また、データアップデートチケット-DUT (PAR) 発行手段 (DUT Issuer) を、パーティションマネージャが兼ねる構成においては、データアップデートチケット-DUT (PAR) 発行手段 (DUT Issuer) のコードは、パーティションマネージャコード (PMC) として設定することが可能である。

【0249】デバイス (Device) の相互認証のタイプ (公開鍵認証、または、共通鍵認証、または、いずれでも可 (Any)) を記録した [Authentication Type] には、チケットを使用した相互認証として実行すべき認証タイプが記録される。具体的には、後段で詳細に説明するが、デバイス認証、パーティション認証のいずれか、または両方の認証を実行する指定、また公開鍵方式、共通鍵方式のどちらを実行するか、またはいずれの認証でも可能であるかについての情報が記録される。

【0250】チケット (Ticket) の正当性検証値 (公開鍵方式 : 署名 (Signature)、共通鍵方式 : MAC) を記録する [Integrity Check Value] フィールドには、公開鍵方式であれば、デバイスアップデートチケット発行手段 (DUT Issuer) の秘密鍵に基づく署名 (図12参照) が生成され格納される。デバイスマネージャ自体がデバイスアップデート登録チケット発行手段 (DU

T Issuer)を兼ねる場合は、デバイスマネージャの秘密鍵を用いて署名が生成される。また、パーティションマネージャ自体がデバイスアップデート登録チケット発行手段(DUT Issuer)を兼ねる場合は、パーティションマネージャの秘密鍵を用いて署名が生成される。この場合、署名検証処理(図13参照)の際は、デバイスマネージャまたはパーティションマネージャの公開鍵が利用される。従って、チケット検証を実行するデバイスは、チケット受領に際し、または前もってデバイスアップデートチケット発行手段(DUT issuer)(ex. デバイスマネージャまたはパーティションマネージャ)の公開鍵(公開鍵証明書)を取得することが必要である。

【0251】デバイスアップデートチケット(DUT)発行手段(DUT Issuer)の公開鍵証明書(CERT\_DUTI)の検証の後、公開鍵証明書(CERT\_DUTI)から取り出したデータアップデートチケット(DUT)発行手段(DUT Issuer)の公開鍵によりICV(Integrity Check Value)の署名検証が可能となる。

【0252】データアップデートチケット(DUT: Data Update Ticket)を適用して更新されるデータ例を図33に示す。

【0253】図33に示すように更新対象データには、デバイスマネージャコード、デバイスマネージャコードバージョン、パーティションマネージャコード、パーティションマネージャコードバージョン、各チケット発行手段コード、各チケットのMAC生成鍵およびバージョン、リポケーションリストなどが含まれる。これら更新対象の各データがデータアップデートチケット(DUT: Data Update Ticket)を適用して、DUTに記録されたルールに従って更新される。更新処理の具体的な手順については、後段でフローを用いて説明する。なお、デバイスマネージャコードバージョン、パーティションマネージャコードバージョン他のバージョン情報は、各バージョンの付加されたデータの更新処理の際に併せて更新されることになる。これらのバージョン情報はデータアップデートチケット(DUT: Data Update Ticket)に格納される。

【0254】[B. ユーザに対するデバイスの配布、デバイスに対する各種設定、デバイス利用処理の詳細についての説明]次に、上述したパーティション分割されたメモリ領域を持つデバイスの利用に至るまでの処理、さらにデバイスの利用処理の詳細についてフローチャート他の図面を参照しながら説明する。説明の手順は以下の項目に従って行なう。

【0255】B1. デバイス初期登録から利用までの流れ

B2. デバイス製造エンティティによる初期登録処理

B3. デバイスマネージャの管轄処理

B3. 1. デバイスマネージャによるデバイス登録処理

B3. 2. デバイスマネージャ管理下における公開鍵証明書発行処理

B4. パーティションマネージャの管轄処理

B4. 1. パーティションマネージャ管理下におけるパーティション登録チケット(PRT)を利用したパーティション設定登録、削除処理

B4. 2. パーティションマネージャ管理下における公開鍵証明書発行処理

B4. 3. ファイル登録チケット(FRT)を利用したファイル生成、消去処理

B5. サービス許可チケット(SPT)を利用したサービス(ファイルアクセス)処理

B6. データアップデートチケット(DUT)を利用したデバイスのデータ更新処理

【0256】[B1. デバイス初期登録から利用までの流れ]EEPROM(フラッシュメモリ)を有するデバイスは、デバイス製造エンティティ(manufacturer)によって製造され、デバイスマネージャによる初期データの書き込みが実行され、ユーザに提供(ex. 販売、貸与)され利用されることになる。ユーザが様々なサービス主体からデバイスを利用したサービスを受けるためには、デバイスのメモリ部にパーティションマネージャによるパーティションが設定され、設定されたパーティション内にサービス提供用のデータを格納したファイルが設定される必要がある。

【0257】また、デバイスに対する様々な処理、すなわちパーティション登録チケット(PRT)を利用したパーティションの設定、ファイル登録チケット(FRT)を利用したファイル設定、さらにサービス許可チケット(SPT)を利用したデータアクセスなどの様々な処理の際に、デバイスとデバイスに対して処理を実行するチケットユーザ(ex. デバイスアクセス機器としてのリーダライタ)との間で様々な手続きが実行される。例えば双方が正当な機器、デバイスであることを確認する相互認証処理、あるいは転送データの正当性を保証し確認するための署名生成、検証処理、さらにデータ暗号化、復号処理などである。本発明の構成では、これらの処理に際して公開鍵証明書を用いた構成を提案している。従って、デバイスによるサービスの利用の前にデバイスに対する公開鍵証明書の発行処理、デバイス格納処理を実行する。

【0258】例えばデバイスとデバイスに対して処理を実行するチケットユーザ(ex. デバイスアクセス機器としてのリーダライタ)との間で公開鍵証明書を用いた相互認証処理が実行され、双方の正当性が確認されたことを条件としてパーティション登録チケット(PRT)を利用したパーティションの設定、ファイル登録チケット(FRT)を利用したファイル設定、さらにサービス許可チケット(SPT)を利用したデータアクセスなどの様々な処理が実行される。また、相互に転送されるデ

ータには必要に応じて電子署名が付加され、検証が実行される。また転送データの暗号化、復号処理も必要に応じて実行されることになる。

【0259】図34は、デバイスの製造から、利用に至るまでの流れを概略的に示した図である。これらの各処理については、フローを参照して後段で詳細に説明するが、全体的な処理の理解のために、図34に示す各段階について簡単に説明する。

【0260】1. まず、デバイスは製造エンティティ (manufacturer) によって製造される。デバイスの製造時には、各デバイスの識別データ (ID) としてのデバイスコードが各デバイスに付与される。デバイスには製造段階で、デバイスコード、製造コードなど、様々の製造情報 (Manufacture Information Block (図14参照)) が書き込まれデバイスのメモリに格納される。

【0261】2. 次に、デバイスマネージャはユーザに対するデバイスの提供前に、自己のID、認証局の公開鍵 (PUB CA (DEV)) など、デバイス管理情報 (Device Management Information (図15参照))、デバイス鍵 (Device Key (図18参照)) などの情報をメモリに格納する。

【0262】3. デバイスマネージャによる管理情報が書き込まれたデバイスは、ユーザに提供される。

【0263】4. 次にユーザは、デバイス対応の公開鍵証明書の取得処理を実行し、取得したデバイス対応公開鍵証明書 (CERT DEV) をデバイスのデバイス鍵領域 (図18参照) に格納する。

【0264】5. デバイスのメモリ部にパーティションを設定し、サービスを提供しようとするサービス主体 (パーティションマネージャ) は、パーティションの設定をデバイスマネージャに要求し、承諾を受けるとともにパーティション登録チケット (PRT) を受領する。また、デバイスとの通信処理において使用する認証局の公開鍵 (PUB CA (PAR)) を指定する。

【0265】6. デバイスは、パーティションマネージャの管理するチケットユーザ (ex. デバイスアクセス機器としてのリーダーライター) との間で通信を実行し、パーティション登録チケット (PRT) を適用したパーティションの登録処理を行なうとともに、認証局の公開鍵 (PUB CA (PAR)) をパーティション鍵領域 (図23参照) 格納する。

【0266】7. パーティションの設定されたデバイスは、パーティション対応公開鍵証明書の発行要求をパーティションマネージャに送信し、取得したパーティション対応公開鍵証明書 (CERT PAR) をパーティション鍵領域 (図23参照) に格納する。

【0267】上記5～7のパーティション設定他の処理は、パーティションを設定してサービスを提供しようとするパーティションマネージャ各々について実行され、複数のパーティションがデバイスに登録される。

【0268】8. 次に、パーティションマネージャは、デバイスに設定したパーティション内に、例えばサービス対応のファイルの設定登録処理をファイル登録チケット (FRT) を適用して実行する。

【0269】9. 10. 設定されたパーティション内にファイルが登録されることにより、例えば電子マネー、定期券などファイル内データによって定義される各種のサービスが実行可能となる。ファイル内のデータ読み取り、データ書き込みなどの処理には、サービス許可チケット (SPT) を適用する。すなわち正当なチケット発行手段が発行したサービス許可チケット (SPT) を適用した場合に限り、SPTに記録されたルールに従ってデータの読み取り、書き込みなどが実行される。

【0270】また、図には示されていないが、必要に応じてデータアップデートチケット (DUT) を使用してデバイスの格納データ中の更新処理対象データ (ex. デバイスマネージャコード、デバイスマネージャコードバージョン、パーティションマネージャコード、パーティションマネージャコードバージョン、各チケット発行手段コード、各チケットのMAC生成鍵およびバージョン、リボケーションリストなど) の更新処理が実行される。なお、デバイスマネージャコードバージョン、パーティションマネージャコードバージョン他のバージョン情報は、各バージョンの付加されたデータの更新処理の際に併せて更新されることになる。これらのバージョン情報はデータアップデートチケット (DUT : Data Update Ticket) に格納される。

【0271】以下、各処理の詳細について、フロー、その他の図を参照しながら説明する。

【0272】[B2. デバイス製造エンティティによる初期登録処理] まず、デバイス製造エンティティによる初期登録処理について、図35を用いて説明する。図35の左側がデバイス製造エンティティ (Manufacture) の登録装置の処理、右側がデバイス (図5参照) の処理を示す。なお、デバイス製造エンティティ (Manufacture) の登録装置は、デバイスに対するデータ読み取り書き込み処理可能な専用のデバイスアクセス機器としてのリーダーライター (図10参照) として構成される。

【0273】まず、ステップS101において登録装置は、デバイスに対して製造情報ブロック (MIB : Manufacture Information Block (図14参照)) の書き込みフラグ (Writable Flag) の読み出しコマンドを送信する。デバイスはコマンドを受信 (S121) すると、デバイスのメモリ部の製造情報ブロック (MIB) 内の書き込み (Writable) フラグを登録装置に送信 (S122) する。

【0274】製造情報ブロック (MIB) 内の書き込み (Writable) フラグを受信 (S102) した登録装置は、書き込みフラグ (Writable Flag) が書き込み可能 (O x f f f f) に設定されているか否かを判別 (S1

03) する。書き込みフラグ (Writable Flag) が書き込み可 (0 x f f f f) に設定されていない場合は、以下の製造情報ブロック (M I B : Manufacture Information Block) の書き込み処理は実行できず、エラーとして終了する。

【0275】書き込みフラグ (Writable Flag) が書き込み可 (0 x f f f f) に設定されている場合は、デバイスの製造情報ブロック (M I B : Manufacture Information Block (図14参照)) を生成 (S104) して M I B 書き込みコマンドとともに、M I B データをデバイスに送信 (S105) する。

【0276】M I B 書き込みコマンド、および M I B データを受信 (S123) したデバイスは、M I B 書き込みフラグ (Writable Flag) を検証 (S124) し、書き込みフラグ (Writable Flag) が書き込み可 (0 x f f f f) に設定されていない場合は、以下の製造情報ブロック (M I B : Manufacture Information Block) の書き込み処理は実行できず、エラーとして終了する。書き込みフラグ (Writable Flag) が書き込み可 (0 x f f f f) に設定されている場合は、受信した M I B データを M I B 領域に書き込む (S125)。

【0277】M I B データ書き込み処理が終了すると書き込み終了通知を登録装置に送信 (S126) する。書き込み終了通知を受信 (S106) した登録装置は初期登録完了コマンドをデバイスに送信 (S107) し、初期登録完了コマンドを受信 (S127) したデバイスは製造情報ブロック (M I B : Manufacture Information Block) の書き込みフラグ (Writable Flag) を書き込み不可 (0 x 0 0 0 0) にセット (S128) し、書き込み終了通知を登録装置に送信 (S129) する。

【0278】書き込み終了通知を受信 (S108) した登録装置は、デバイスに対して製造情報ブロック (M I B : Manufacture Information Block (図14参照)) の書き込みフラグ (Writable Flag) の読み出しコマンドを送信 (S109) する。デバイスはコマンドを受信 (S130) すると、デバイスのメモリ部の製造情報ブロック (M I B) 内の書き込みフラグ (Writable Flag) を登録装置に送信 (S131) する。

【0279】製造情報ブロック (M I B) 内の書き込みフラグ (Writable Flag) を受信 (S110) した登録装置は、書き込みフラグ (Writable Flag) が書き込み不可 (0 x 0 0 0 0) に設定されているか否かを判別 (S111) する。書き込みフラグ (Writable Flag) が書き込み不可 (0 x 0 0 0 0) に設定されていない場合は、正常な M I B データ書き込み処理が終了していないことを示し、エラーとして処理を終了する。書き込みフラグ (Writable Flag) が書き込み不可 (0 x 0 0 0 0) に設定されている場合は、正常な M I B データ書き込み処理が終了したものとして処理を終了する。

【0280】[B3. デバイスマネージャの管轄処理]

次に、デバイスマネージャの管轄処理について説明する。ここでは、デバイスの使用開始以前に実行される処理について説明する。デバイスの使用開始以前に実行されるデバイスマネージャの処理としては、デバイスのメモリ部のデバイス管理情報ブロック (D M I B : Device Management Information Block)、公開鍵系デバイスキー定義ブロック (D K D B : Device Key Definition Block (PUB))、共通鍵系デバイスキー定義ブロック (D K D B : Device Key Definition Block (Common))、デバイス鍵領域 (Device Key Area) に対するデータ書き込み処理として実行するデバイス登録処理と、デバイスに対してデバイス対応公開鍵証明書 (CERT DEV) を発行する処理がある。以下、これらの処理の詳細について説明する。

【0281】[B3. 1. デバイスマネージャによるデバイス登録処理] 図36以下のフローを用いて、デバイスマネージャによるデバイスに対するデバイス管理情報他の格納処理を伴う初期登録処理について説明する。図36以下のフローにおいて、左側がデバイスマネージャ (DM) の初期登録装置の処理、右側がデバイス (図5参照) の処理を示す。なお、デバイスマネージャ (DM) の初期登録装置は、デバイスに対するデータ読み取り書き込み処理可能な装置 (ex. デバイスアクセス機器としてのリーダライタ、PC) であり、図10のデバイスアクセス機器としてのリーダライタに相当する構成を有する。

【0282】まず、ステップS201において、デバイスの識別子 I D m の読み出し (Read) コマンドをデバイスに出力する。デバイスはコマンドを受信 (S211) し、デバイスの識別子 I D m を登録装置に送信 (S212) する。

【0283】デバイスの識別子 I D m を受信 (S202) した登録装置は、ステップS203において、デバイスに対してデバイス管理情報ブロック (D M I B : Device Management Information Block (図15参照)) の書き込みフラグ (Writable Flag) の読み出しコマンドを送信する。デバイスはコマンドを受信 (S213) すると、デバイスのメモリ部のデバイス管理情報ブロック (D M I B) 内の書き込みフラグ (Writable Flag) を登録装置に送信 (S214) する。

【0284】デバイス管理情報ブロック (D M I B) 内の書き込みフラグ (Writable Flag) を受信 (S204) した登録装置は、書き込みフラグ (Writable Flag) が書き込み可 (0 x f f f f) に設定されているか否かを判別 (S205) する。書き込みフラグ (Writable Flag) が書き込み可 (0 x f f f f) に設定されていない場合は、以下のデバイス管理情報ブロック (D M I B : Device Management Information Block) の書き込み処理は実行できず、エラーとして終了する。

【0285】書き込みフラグ (Writable Flag) が書き

込み可 (0 x f f f f) に設定されている場合は、デバイスマネージャコード (DMC) および DMC バージョンの書き込み (DMC Write) コマンドをデバイスに送信 (S 2 0 6) する。このコードは、コード管理機関 (図 1 ~ 図 3 参照) によりデバイスマネージャに対して予め割り当てられたデータである。

【0 2 8 6】DMC Write コマンドを受信 (S 2 1 5) したデバイスは、DMI B 書き込みフラグ (Writable Flag) を検証 (S 2 1 6) し、書き込みフラグ (Writable Flag) が書き込み可 (0 x f f f f) に設定されていない場合は、以下のデバイス管理情報ブロック (DMI B : Device Management Information Block) の書き込み処理は実行できず、エラーとして終了する。書き込みフラグ (Writable Flag) が書き込み可 (0 x f f f f) に設定されている場合は、受信したデバイスマネージャコード (DMC) および DMC バージョンを DMI B 領域に書き込む (S 2 1 7)。

【0 2 8 7】デバイスマネージャコード (DMC) および DMC バージョンの書き込み処理が終了すると書き込み終了通知を登録装置に送信 (S 2 1 8) する。書き込み終了通知を受信 (S 2 0 7) した登録装置は、次にデバイス総ブロック数 (Device Total Block Number) 書き込みコマンドをデバイスに送信 (S 2 0 8) する。

【0 2 8 8】デバイス総ブロック数 (Device Total Block Number) 書き込みコマンドを受信 (S 2 1 9) したデバイスは、DMI B 書き込みフラグ (Writable Flag) を検証 (S 2 2 0) し、書き込みフラグ (Writable Flag) が書き込み可 (0 x f f f f) に設定されていない場合は、以下のデバイス管理情報ブロック (DMI B : Device Management Information Block) の書き込み処理は実行できず、エラーとして終了する。書き込みフラグ (Writable Flag) が書き込み可 (0 x f f f f) に設定されている場合は、受信したデバイス総ブロック数 (Device Total Block Number) を DMI B 領域に書き込む (S 2 2 1)。さらに、デバイスは、DMI B 領域のデバイス空きブロック数情報領域 (Free Block Number in Device) に TB-4 を書き込む (S 2 2 2)。TB はデバイス総ブロック数 (Device Total Block Number) を意味する。なお TB-4 の 4 ブロックは、製造情報ブロック (MIB : Manufacture Information Block)、デバイス管理情報ブロック (DMI B : Device Management Information Block)、公開鍵系デバイスキー定義ブロック (DKDB : Device Key Definition Block (PUB))、共通鍵系デバイスキー定義ブロック (DKDB : Device Key Definition Block (Common)) を示している。

【0 2 8 9】次に、デバイスは、デバイス管理情報ブロック (DMI B) のパーティション数 (Partition Number) 領域に 0 を書き込む (S 2 2 3)。この時点でデバイスにはパーティションは設定されていないからであ

る。さらに DMI B の空き領域のポインタ (Pointer of Free Area) に 0 を書き込み (S 2 2 4)、書き込み処理完了を登録装置に送信 (S 2 2 5) する。

【0 2 9 0】書き込み処理完了通知をデバイスから受信 (S 2 0 9) した登録装置は、次に、デバイス認証に共通鍵を用いるか否かを判定 (S 2 3 1) する。認証処理については、後段で詳細に説明するが、公開鍵認証方式、共通鍵認証方式のいずれかを実行する構成が可能であり、デバイスマネージャは、デバイスに必要な認証方式を設定することが可能となる。デバイスが共通鍵認証を実行するデバイスであれば、デバイスマネージャは共通鍵認証に必要な情報 (e x. 認証鍵生成用のマスター鍵他) をデバイスにセットし、デバイスが共通鍵認証を実行しないデバイスであれば、これらの情報をデバイスに格納しないことになる。デバイスマネージャは、デバイスの採用する認証方式に応じて共通鍵認証、公開鍵認証のいずれか、あるいは両方式を実行可能なデータをデバイスに設定する。

【0 2 9 1】図 3 7 に示すように、デバイス認証に共通鍵を用いる場合、ステップ S 2 3 2 ~ S 2 3 3、S 2 4 1 ~ S 2 4 5 を実行し、デバイス認証に共通鍵を用いない場合、これらのステップは省略される。

【0 2 9 2】デバイス認証に共通鍵を用いる場合、ステップ S 2 3 2 において登録装置は、共通鍵認証データ書き込みコマンドとして、MKauth\_DEV\_A : 双方向個別鍵認証用マスター鍵、Kauth\_DEV\_B : 双方向個別鍵認証用共通鍵、IRL\_DEV : 排除デバイス (Device) のデバイス識別子 (ID) を登録したリボケーションリスト (Revocation List (Device ID))、およびこれらのバージョン情報をデバイスに送信する。

【0 2 9 3】ステップ S 2 4 1 でデバイスは、上述の書き込みコマンドを受信し、ステップ S 2 4 2 において、DMI B の書き込みフラグ (Writable Flag) が書き込み可であることを確認して受領データをデバイス鍵領域 (図 1 8 参照) に書き込む (S 2 4 3)。次にデータ書き込みによって生じたポインタ、サイズ、デバイス内のフリーブロック数の調整を実行 (S 2 4 4) し、書き込み終了通知を登録装置に送信 (S 2 4 5) する。

【0 2 9 4】書き込み終了通知を受信 (S 2 3 3) した登録装置は、ステップ S 2 3 4 においてデバイス認証に公開鍵を用いるか否かを判定する。図 3 7 に示すように、デバイス認証に公開鍵を用いる場合、ステップ S 2 3 5 ~ S 2 3 9、S 2 4 6 ~ S 2 5 4 を実行し、デバイス認証に公開鍵を用いない場合、これらのステップは省略される。

【0 2 9 5】デバイス認証に公開鍵を用いる場合、ステップ S 2 3 5 において登録装置は、公開鍵認証データ書き込みコマンドとして、PUB\_CA (DEV) : デバイスマネージャ対応公開鍵を発行する認証局 CA (DEV) の公開鍵、PARAM\_DEV : デバイス (Device) の公開鍵パラメー

タ、CRL\_DEV : 排除デバイス (Device) の公開鍵証明書識別子 (ex. シリアルナンバ: SN) を登録したりボケーションリスト (Revocation List (Certificate)、およびこれらのバージョン情報をデバイスに送信する。

【0296】ステップS246でデバイスは、上述の書き込みコマンドを受信し、ステップS247において、DMIBの書き込みフラグ (Writable Flag) が書き込み可であることを確認して受領データをデバイス鍵領域 (図18参照) に書き込む (S248)。次にデータ書き込みによって生じたポインタ、サイズ、デバイス内のフリーブロック数の調整を実行 (S249) し、書き込み終了通知を登録装置に送信 (S250) する。

【0297】書き込み終了通知を受信 (S236) した登録装置は、公開鍵と秘密鍵の鍵ペア生成コマンドをデバイスに送信 (S237) する。なお、この実施例では、鍵ペアの生成はデバイスが実行する構成としているが、例えば登録装置が実行してデバイスに提供する構成としてもよい。

【0298】鍵ペア生成コマンドを受信 (S251) したデバイスは、デバイス内の暗号処理部 (図5参照) において公開鍵 (PUB DEV) と秘密鍵 (PRI DEV) のペアを生成し、生成した鍵をデバイス鍵領域 (図18参照) に書き込む (S252)。なお、公開鍵 (PUB DEV) については、デバイス鍵領域のCERT・DEV領域に一時格納し、その後、公開鍵 (PUB DEV) を格納した公開鍵証明書を受領した時点で公開鍵証明書 (CERT) に置き換えられる。次にデータ書き込みによって生じたポインタ、サイズ、デバイス内のフリーブロック数の調整を実行 (S253) し、生成格納した公開鍵を登録装置に送信 (S254) する。

【0299】登録装置は、デバイスから公開鍵 (PUB DEV) を受信し、先にデバイスから受信したデバイスの識別子IDmとともに、デバイスマネージャ内のデータベース (DB (DEV) (図7参照)) に保存する。

【0300】次に、デバイスマネージャの登録装置は、パーティション登録チケット (PRT: Partition Registration Ticket) の検証処理に共通鍵を用いるか否かを判定 (S261) する。チケット検証には、後段で詳細に説明するがMAC値検証等による共通鍵方式と、前述の図12、図13を用いて説明した秘密鍵による署名生成、公開鍵による署名検証を行なう公開鍵方式のいずれかを適用することが可能であり、デバイスマネージャは、デバイスの採用する検証処理方式を設定することができる。デバイスマネージャは、デバイスの採用するPRTチケット検証方式に応じて共通鍵、公開鍵のいずれか、あるいは両方式を実行可能なデータをデバイスに設定する。

【0301】デバイスマネージャは、デバイスが共通鍵認証を実行するデバイスであれば、デバイスマネージャ

は共通鍵方式のPRT検証に必要な情報 (ex. PRT検証共通鍵) をデバイスにセットし、デバイスが共通鍵認証を実行しないデバイスであれば、これらの情報をデバイスに格納しないことになる。

【0302】図38に示すように、PRT検証に共通鍵方式を用いる場合、ステップS262~263、S271~S275を実行し、PRT検証に共通鍵を用いない場合、これらのステップは省略される。

【0303】PRT検証に共通鍵を用いる場合、ステップS262において登録装置は、PRT検証共通鍵書き込みコマンドとして、Kprt: パーティション登録チケット (PRT) のMAC検証用鍵、およびバージョン情報をデバイスに送信する。

【0304】ステップS271でデバイスは、上述の書き込みコマンドを受信し、ステップS272において、DMIBの書き込みフラグ (Writable Flag) が書き込み可であることを確認して受領データをデバイス鍵領域 (図18参照) に書き込む (S273)。次にデータ書き込みによって生じたポインタ、サイズ、デバイス内のフリーブロック数の調整を実行 (S274) し、書き込み終了通知を登録装置に送信 (S275) する。

【0305】書き込み終了通知を受信 (S263) した登録装置は、ステップS264においてPRT検証に公開鍵を用いるか否かを判定する。図38に示すように、PRT検証に公開鍵を用いる場合、ステップS265~S266、S276~S282を実行し、PRT検証に公開鍵を用いない場合、これらのステップは省略される。

【0306】PRT検証に公開鍵を用いる場合、ステップS265において登録装置は、PRT検証データ書き込みコマンドとして、PRTIC (PRT Issuer Category): パーティション登録チケット (PRT) 発行者カテゴリ、PUB\_CA (DEV): デバイスマネージャ対応公開鍵を発行する認証局CA (DEV) の公開鍵、PARAM\_DEV: デバイス (Device) の公開鍵パラメータ、CRL\_DEV: 排除デバイス (Device) の公開鍵証明書識別子 (ex. シリアルナンバ: SN) を登録したりボケーションリスト (Revocation List (Certificate)、およびこれらのバージョン情報をデバイスに送信する。

【0307】ステップS276でデバイスは、上述の書き込みコマンドを受信し、ステップS277において、DMIBの書き込みフラグ (Writable Flag) が書き込み可であることを確認して、ステップS278において、受領データ中のPRTIC (PRT Issuer Category): パーティション登録チケット (PRT) 発行者カテゴリを公開鍵系デバイス鍵定義ブロック (DKDB: Device Key Definition block (PUB) (図16参照)) に書き込みバージョン情報を同ブロックのバージョン領域に書き込む。

【0308】次にデバイスは、ステップS279におい



て、PUB\_CA(DEV) : デバイスマネージャ対応公開鍵を発行する認証局CA (DEV) の公開鍵データが書き込み済みか否かを判定し、書き込まれていない場合にステップS280において、PUB\_CA(DEV)、PARAM\_DEV、CRL\_DEVをデバイス鍵領域(図18参照)に書き込む。次にデータ書き込みによって生じたポインタ、サイズ、デバイス内のフリーブロック数の調整を実行(S281)し、書き込み終了通知を登録装置に送信(S282)する。

【0309】書き込み終了通知を受信(S266)した登録装置は、次に、ステップS291において、共通鍵データの更新をサポートするデバイスであるか否かを判定する。デバイスに格納されたデータ中、そのいくつかは更新対象データとして前述したデータアップデートチケット(DUT: Data Update Ticket)(図32参照)を用いて更新が可能である。更新対象となるデータは、先に図33を用いて説明した通りである。このデータアップデートチケット(DUT: Data Update Ticket)を用いた更新処理においても共通鍵方式、または公開鍵方式のいずれかの方式が可能であり、デバイスマネージャはデバイスに応じていずれかの方式または両方式を実行可能なデータをデバイスに設定する。

【0310】デバイスマネージャは、デバイスが共通鍵方式によるデータ更新を実行するデバイスであれば、共通鍵方式のデータ更新処理に必要な情報(ex. データアップデートチケット(DUT)のMAC検証用鍵他)をデバイスにセットし、デバイスが共通鍵認証を実行しないデバイスであれば、これらの情報をデバイスに格納しないことになる。

【0311】図39に示すように、データアップデートチケット(DUT: Data Update Ticket)を用いたデータ更新処理に共通鍵方式を用いる場合、ステップS292～S293、S301～S305を実行し、データ更新に共通鍵方式を用いない場合、これらのステップは省略される。

【0312】データ更新に共通鍵を用いる場合、ステップS292において登録装置は、データアップデートチケット(DUT: Data Update Ticket)検証共通鍵書き込みコマンドとして、Kdut\_DEV1: データアップデートチケット(DUT)のMAC検証用鍵、Kdut\_DEV2: データ更新用暗号鍵、Kdut\_DEV3: データアップデートチケット(DUT)のMAC検証用鍵、Kdut\_DEV4: データ更新用暗号鍵およびこれらのバージョン情報をデバイスに送信する。

【0313】ステップS301でデバイスは、上述の書き込みコマンドを受信し、ステップS302において、DMIBの書き込みフラグ(Writable Flag)が書き込み可であることを確認して受領データをデバイス鍵領域(図18参照)に書き込む(S303)。次にデータ書き込みによって生じたポインタ、サイズ、デバイス内のフリーブロック数の調整を実行(S304)し、書き込

み終了通知を登録装置に送信(S305)する。

【0314】書き込み終了通知を受信(S293)した登録装置は、ステップS294において、デバイスが公開鍵方式を用いたデータアップデートチケット(DUT: Data Update Ticket)を使用したデータ更新処理をサポートするか否かを判定する。図39に示すように、公開鍵方式をサポートする場合、ステップS295～S296、S306～S310を実行し、公開鍵方式をサポートしない場合、これらのステップは省略される。

【0315】公開鍵方式をサポートする場合、ステップS295において登録装置は、データアップデートチケット(DUT: Data Update Ticket)発行者コード書き込みコマンドとして、DUTIC\_DEV(DUT Issuer Category): データアップデートチケット(DUT: Data Update Ticket)発行者カテゴリ、およびバージョン情報をデバイスに送信する。

【0316】ステップS306でデバイスは、上述の書き込みコマンドを受信し、ステップS307において、DMIBの書き込みフラグ(Writable Flag)が書き込み可であることを確認して、ステップS308において、受領データを公開鍵系デバイス鍵定義ブロック(DKDB(PUB): Device Key Definition Block(PUB))に書き込む(S308)。次にデータ書き込みによって生じたポインタ、サイズ、デバイス内のフリーブロック数の調整を実行(S309)し、書き込み終了通知を登録装置に送信(S310)する。

【0317】書き込み終了通知を受信(S296)した登録装置は、次にステップS321において、デバイスマネージャ(DM)初期登録完了コマンドをデバイスに対して送信する。コマンドを受領(S331)したデバイスは、ステップS332において、相互認証、パーティション登録チケット(PRT)の検証、さらにデータアップデートチケット(DUT)の検証、それぞれについて少なくとも公開鍵方式、共通鍵方式のいずれかの処理が実行可能なデータが設定済みであるか否かを判定する。これらのデータに不足がある場合は、いずれかの処理が実行できないことになり、デバイスマネージャによる初期登録はエラーと判定され処理を終了する。

【0318】ステップS332において、相互認証、パーティション登録チケット(PRT)の検証、さらにデータアップデートチケット(DUT)の検証、それぞれについて少なくとも公開鍵方式、共通鍵方式のいずれかの処理が実行可能なデータが設定済みであると判定した場合は、ステップS333においてデバイスは、デバイス管理情報ブロック(DMIB: Device Management Information Block)の書き込み(Writable)フラグを書き込み不可(0x0000)にセットし、書き込み終了通知を登録装置に送信(S334)する。

【0319】書き込み終了通知を受信(S322)した登録装置は、デバイスに対してデバイス管理情報ブロッ

ク (DMIB : Device Management Information Block) (図15参照) の書き込みフラグ (Writable Flag) の読み出しコマンドを送信 (S323) する。デバイスはコマンドを受信 (S335) すると、デバイスのメモリ部のデバイス管理情報ブロック (DMIB) 内の書き込みフラグ (Writable Flag) を登録装置に送信 (S336) する。

【0320】デバイス管理情報ブロック (DMIB) 内の書き込みフラグ (Writable Flag) を受信 (S324) した登録装置は、書き込みフラグ (Writable Flag) が書き込み不可 (0x0000) に設定されているか否かを判別する。書き込みフラグ (Writable Flag) が書き込み不可 (0x0000) に設定されていない場合は、正常なDMIBデータ書き込み処理が終了していないことを示し、エラーとして処理を終了する。書き込みフラグ (Writable Flag) が書き込み不可 (0x0000) に設定されている場合は、正常なDMIBデータ書き込み処理が終了したものとして処理を終了する。

【0321】デバイス製造エンティティ (Manufacture) の登録装置による初期登録 (図35の処理フロー) および、デバイスマネージャによる初期登録処理 (図36～図40の処理フロー) が完了した状態のデバイスのメモリ内格納データ構成例を図41に示す。図41は、図6、図14乃至図18を用いて説明した製造情報ブロック (Manufacture Information Block)、デバイス管理情報ブロック (Device Management Information Block)、公開鍵系デバイス鍵定義 (Device Key Definition Block (PUB))、共通鍵系デバイス鍵定義ブロック (Device Key Definition Block (Common))、デバイス鍵領域 (Device Key Area) を示すものである。この時点では、メモリにパーティションは形成されていない。

【0322】製造情報ブロック (Manufacture Information Block) には、図14を用いて説明したように、デバイスの固有情報としてのデバイスコード他が書き込まれる。この製造情報ブロック (Manufacture Information Block) に書き込まれた情報、あるいは書き込まれた情報の一部、または書き込まれた情報に基づいて取得される演算データがデバイスの識別子 (IDm) に相当する。

【0323】なお、図に示すデバイス鍵領域 (Device Key Area) には Kauth\_DEV\_B : 双方向個別鍵認証用共通鍵、MKauth\_DEV\_A : 双方向個別鍵認証用マスター鍵が格納されているが、これらの鍵は、デバイスが共通鍵認証処理を行なう要請が無い場合は格納しない構成としてもよく、また、Kprt : パーティション登録チケット (PRT) のMAC検証用鍵についても、デバイスが共通鍵によるチケット検証処理を実行しない構成の場合には格納しない構成としてもよい。

【0324】また、IRL\_DEV : 排除デバイス (Device) のデバイス識別子 (ID) を登録したリボケーションリ

スト (Revocation List (Device ID))、CRL\_DEV : 排除デバイス (Device) の公開鍵証明書識別子 (ex. シリアルナンバ : SN) を登録したリボケーションリスト (Revocation List (Certificate)) についても、デバイス発行時点でリボーク (排除) されたデバイスが存在しない場合、あるいは他のソースを使用してリボケーションリストを取得する構成とする場合には、リボケーションリストを格納しない構成としてもよい。

【0325】【B3. 2. デバイスマネージャ管理下における公開鍵証明書発行処理】次に図42以下を用いて、デバイスマネージャによるデバイス対応公開鍵証明書の発行処理について説明する。デバイスには、デバイス全体の認証、デバイスを単位とした処理に適用可能なデバイス対応公開鍵証明書 (CERT DEV) と、デバイス内の特定のパーティションに対する処理の際の認証その他検証処理等に適用可能なパーティション対応公開鍵証明書 (CERT PAR) が格納され得る。パーティション対応公開鍵証明書 (CERT PAR) は、デバイスに設定されたパーティション毎に設定格納可能である。

【0326】デバイス対応公開鍵証明書 (CERT DEV) は、デバイスマネージャの管轄するメモリ領域であるデバイス鍵領域 (Device Key Area) (図18参照) に格納され、パーティション対応公開鍵証明書 (CERT PAR) は、各パーティションマネージャの管轄するメモリ領域であるパーティション鍵領域 (Partition Key Area) (図23参照) に格納される。

【0327】デバイス対応公開鍵証明書 (CERT DEV) は、デバイスマネージャの管轄する登録局を介して認証局 (CA for DM) (図2、図3参照) の発行した公開鍵証明書をデバイスに付与する手続きにより発行され、デバイスマネージャの管轄登録局が発行した公開鍵証明書 (CERT DEV) についての管理 (データベース222 (図7参照)) を実行する。

【0328】またパーティション対応公開鍵証明書 (CERT PAR) は、パーティションマネージャの管轄する登録局を介して認証局 (CA for PM) (図2、図3参照) の発行した公開鍵証明書をデバイスに付与する手続きにより発行され、パーティションマネージャの管轄登録局が発行した公開鍵証明書 (CERT PAR) についての管理 (データベース332 (図9参照)) を実行する。

【0329】図42および図43に従って、デバイスマネージャの管轄登録局によるデバイスに対するデバイス対応公開鍵証明書 (CERT DEV) の発行処理の手順を説明する。なお、デバイスマネージャの登録局 (RA) 構成のみを取り出した発行装置 (DM)、認証局 (CA)、ユーザデバイスとの関係を図44に示した。図44に示すように制御手段221は暗号処理手段を有する。なお暗号処理は暗号処理に関するプログラムを制

御部（CPU（図8の2111）の制御下において実行することによって行われる。

【0330】図42、図43において、左側がデバイスマネージャの管轄登録局のCERT（公開鍵証明書）発行装置、具体的には、図7に示すデバイスマネージャの構成図における制御手段221の処理、右側がデバイスの処理である。

【0331】まずステップS351において、CERT発行装置は、デバイス対応公開鍵証明書（CERT DEV）の発行対象となるデバイスのユーザ情報を取得し、証明書発行の許可（判定）を行ない発行対象となるデバイスとの通信路を確保する。デバイス対応公開鍵証明書（CERT DEV）の発行対象となるデバイスのユーザ情報は、例えばデバイスの初期登録時に生成したデータから取得可能である。また、別途、別経路にてユーザの名前や住所、電話番号、e-mailアドレスなどを取得するようにしてもよい。なお、ユーザ情報はデバイスとの通信路設定後、デバイスから取得してもよい。通信路は、有線、無線を問わずデータ送受信可能な通信路として確保されればよい。

【0332】次にCERT発行装置は、ステップS352において、乱数を含む認証データ生成コマンドをデバイスに対して送信する。認証データ生成コマンドを受信（S361）したデバイスは、受信乱数Rと、デバイス識別子（IDm）の結合データにデバイス秘密鍵（PRI DEV）を適用してデジタル署名（S）の生成処理（図12参照）を実行（S362）する。デバイスは、デバイスの識別データ（IDm）と署名（S）をCERT発行装置に送信する。

【0333】デバイスから識別データ（IDm）と署名（S）を受信（S353）したCERT発行装置は、受信したデバイス識別データ（IDm）を検索キーとしてデータベースDB（DEV）222から格納済みのデバイス公開鍵（PUB DEV）を取得する。さらに、取得したデバイス公開鍵（PUB DEV）を適用して署名（S）の検証処理（図13参照）を実行（S355）する。検証に成功しなかった場合は、デバイスからの送信データは不正なデータであると判定し処理は終了する。

【0334】検証に成功した場合は、認証局（CA for DM）610に対してデバイス対応公開鍵証明書（CERT DEV）の発行処理を依頼（S357）する。デバイスマネージャは認証局610の発行したデバイス対応公開鍵証明書（CERT DEV）を受信（S358）してデバイスに送信（S359）する。

【0335】デバイスマネージャ（登録局）からデバイス対応公開鍵証明書（CERT DEV）を受信したデバイスは、予めデバイス鍵領域に格納済みの認証局の公開鍵（PUB CA（DEV））を用いて受信したデバイス対応公開鍵証明書（CERT DEV）の署名検証

を実行する。すなわち公開鍵証明書には認証局の秘密鍵で実行された署名があり（図11参照）、この署名検証（S366）を行なう。

【0336】署名検証に失敗した場合は、正当な公開鍵証明書でないと判定し、エラー通知をCERT発行装置に対して実行（S385）する。

【0337】署名検証に成功した場合は、デバイス対応公開鍵証明書（CERT DEV）に格納されたデバイス公開鍵（PUB DEV）と自デバイスに保管されたデバイス公開鍵（PUB DEV）の比較を実行（S382）し、一致しない場合はエラー通知を実行し、一致した場合は、受信したデバイス対応公開鍵証明書（CERT DEV）をデバイス鍵領域（図18参照）に格納（S383）する。なお、デバイス対応公開鍵証明書（CERT DEV）の発行以前は、この領域に自デバイスで生成した公開鍵（PUB DEV）を格納し、正当なデバイス対応公開鍵証明書（CERT DEV）が発行された時点で、デバイス対応公開鍵証明書（CERT DEV）により上書きする処理として格納する。

【0338】デバイス対応公開鍵証明書（CERT DEV）の格納が終了すると格納処理終了通知をCERT発行装置に送信（S384）する。CERT発行装置は、格納処理終了通知を受信（S371）し、格納成功を確認（S372）して処理を終了する。格納成功の確認が得られない場合はエラーとして処理が終了する。

【0339】図45にデバイス対応公開鍵証明書（CERT DEV）の発行処理において、デバイスマネージャ200、デバイス100、認証局（CA）610各エンティティ間のデータ送受信処理について説明した図を示す。

【0340】図45中のNo. 1～14の順に処理が実行される。なお、処理No. 1のデバイスマネージャ200によるデバイス100からのデバイス識別子（IDm）、デバイス公開鍵（PUB DEV）の取得処理、および処理No. 2のデバイス識別子（IDm）の登録処理はデバイスマネージャによる初期登録において実行される処理である。

【0341】デバイス対応公開鍵証明書（CERT DEV）の発行手続きは、処理No. 3からであり、3. デバイスマネージャによるデバイスからの顧客情報取得、4. 顧客情報の登録（登録済みである場合は不要）、5. デバイスからのデバイス識別子（IDm）取得、6. 取得したデバイス識別子（IDm）にもとづいてデータベース検索を実行して対応の公開鍵（PUB DEV）を取得、7. デバイスとデバイスマネージャ間における認証処理、この処理は図42、図43の処理においては省略されていたが、図42、図43においてはデバイスからのデバイス識別子（IDm）取得の際に署名検証を実行しており、通信相手の送信データの確認により、認証を省略した。図42、図43における署名検

証、図45の認証の少なくともいずれか、またはいずれをも実行することが望ましい。なお、認証処理の詳細については、後段のB. 4. パーティションマネージャの管轄処理の項目で説明する。

【0342】8. デバイスマネージャから認証局に対するデバイス対応公開鍵証明書の発行要求、9. デバイス対応公開鍵証明書(CERT DEV)の生成、10. 認証局における生成公開鍵証明書のデータ登録、11. 認証局(CA)610からデバイスマネージャ200に対する公開鍵証明書の配布、12. デバイスマネージャのデータベース更新(公開鍵証明書発行情報登録)、13. デバイスマネージャからデバイスに対するデバイス対応公開鍵証明書(CERT DEV)の送信、14. デバイスにおけるデバイス対応公開鍵証明書(CERT DEV)の保存、保存は、前述したようにデバイス鍵領域に書き込み保存される。

【0343】以上の処理により、デバイスは、デバイス対応公開鍵証明書(CERT DEV)を取得し、メモリ部に格納する。このデバイス対応公開鍵証明書(CERT DEV)をメモリのデバイス鍵格納領域に格納した後のメモリの各ブロックのデータ格納構成を図46に示す。図46は、先に図6、図14乃至図18を用いて説明した製造情報ブロック(Manufacture Information Block)、デバイス管理情報ブロック(Device Management Information Block)、公開鍵系デバイス鍵定義(Device Key Definition Block(PUB))、共通鍵系デバイス鍵定義ブロック(Device Key Definition Block(Common))、デバイス鍵領域(Device Key Area)を示すものである。この時点では、メモリにパーティションは形成されていない。

【0344】図46に示すデバイス鍵領域(Device Key Area)にはデバイス対応公開鍵証明書(CERT DEV)が格納される。デバイス対応公開鍵証明書(CERT DEV)の発行前には、この領域には、デバイスが生成した公開鍵(PUB DEV)が格納され、デバイス対応公開鍵証明書(CERT DEV)を受信すると、デバイス対応公開鍵証明書(CERT DEV)によって上書き処理がなされる。なお、この上書き処理によりポインタ、サイズ、管理データがある場合は必要な変更処理を実行する。

【0345】[B4. パーティションマネージャの管轄処理]次に、パーティションマネージャの管轄処理について説明する。ここでは、デバイスの使用開始以前に実行される処理について説明する。デバイスの使用開始以前に実行されるパーティションマネージャの処理としては、デバイスのメモリ部にパーティションを設定する処理と、デバイスに対してパーティション対応公開鍵証明書(CERT PAR)を発行する処理がある。以下、これらの処理の詳細について説明する。パーティションを設定する処理には、デバイスとパーティションマネー

ジャ間における相互認証処理(デバイス認証またはパーティション認証)、パーティション登録チケット(PRT: Partition Registration Ticket)の正当性検証処理が含まれる。なお、パーティションの削除処理についても基本的にパーティション作成と同様の手続きに従って実行可能であるので併せて説明する。

【0346】[B4. 1. パーティションマネージャ管理下におけるパーティション登録チケット(PRT)を利用したパーティション設定登録、削除処理]まず、パーティション登録チケット(PRT)(図26参照)を利用したパーティション設定登録、削除処理について説明する。図47以下のフロー他の図面を参照して説明する。なお、上述のように、パーティション設定処理には、デバイスとパーティションマネージャ間における相互認証処理(デバイス認証またはパーティション認証)、パーティション登録チケット(PRT: Partition Registration Ticket)の正当性検証処理が含まれ、これらの処理についても説明する。

【0347】図47に示すパーティション設定登録、削除処理フローについて説明する。図47において、左側がパーティションマネージャのパーティション作成・削除装置、右側がデバイス(図5参照)の処理を示す。なお、パーティションマネージャのパーティション作成・削除装置は、デバイスに対するデータ読み取り書き込み処理可能な装置(ex. デバイスアクセス機器としてのリーダライタ、PC)であり、図10のデバイスアクセス機器としてのリーダライタに相当する。まず、図47を用いて、パーティション作成、削除処理の概要を説明し、その後、当処理に含まれる各処理の詳細を図48以下のフローを用いて順次説明する。

【0348】まず、図47のステップS401とS410において、パーティション作成・削除装置とデバイス間での相互認証処理が実行される。データ送受信を実行する2つの手段間では、相互に相手が正しいデータ通信者であるか否かを確認して、その後に必要なデータ転送を行なうことが行われる。相手が正しいデータ通信者であるか否かの確認処理が相互認証処理である。相互認証処理時にセッション鍵の生成を実行して、生成したセッション鍵を共有鍵として暗号化処理を実行してデータ送信を行なう構成が1つの好ましいデータ転送方式である。

【0349】本発明のシステムにおける相互認証処理では、デバイス認証またはパーティション認証のいずれかが実行される。また、それぞれについて共通鍵方式認証、あるいは公開鍵方式認証処理のいずれかが適用される。これらについては後述する。

【0350】なお、相互認証処理として実行すべき処理は、適用するパーティション登録チケット(PRT)

(図26参照)の

\* Authentication Flag : チケット(Ticket)の利用処

理においてデバイス (Device) との相互認証が必要か否かを示すフラグ

\* Authentication Type : デバイス (Device) の相互認証のタイプ (公開鍵認証、または、共通鍵認証、または、いずれでも可 (Any))

によって決定される。

【0351】認証処理に失敗した場合 (S402, S411でNo) は、相互が正当な機器、デバイスであることの確認がとれないことを示し、以下の処理は実行されずエラーとして処理は終了する。

【0352】認証処理に成功すると、パーティション作成・削除装置は、デバイスに対してパーティション登録チケット (PRT: Partition Registration Ticket) を送信する。パーティション登録チケット (PRT) は、パーティションマネージャの要求により、デバイスマネージャの管理するパーティション登録チケット (PRT) 発行手段 (PRT Issuer) によりパーティションマネージャに対して発行されるチケットである。パーティション登録チケット (PRT) は、デバイスに対するアクセス制御チケットであり、先に説明した図26のデータフォーマット構成を持つチケットである。

【0353】なお、パーティション登録チケット (PRT) を、チケットユーザに対して送信する際には、公開鍵方式の場合、パーティション登録チケット (PRT) 発行手段 (PRT Issuer) の公開鍵証明書 (CERT\_PRTI) も一緒に送信する。PRT発行手段の公開鍵証明書 (CERT\_PRTI) の属性 (Attribute) は、パーティション登録チケット (PRT) 発行手段 (PRT User) の識別子 (PRTIC) と一致する。

【0354】パーティション登録チケット (PRT) を受信 (S412) したデバイスは、受信したチケット (PRT) の正当性と利用者チェック処理を実行 (S413) する。チケットの正当性の検証処理は、共通鍵方式によるMAC検証、あるいは公開鍵方式による署名検証処理のいずれかを適用して実行される。利用者チェックは、チケットを送信してきた機器 (チケット利用者) の正当性をチェックする処理であり、相互認証が成立済みであり、認証相手の識別データと、チケットに記録されているチケットユーザ識別子 (図26参照) との一致等を検証する処理として実行される。これらの処理の詳細については後述する。

【0355】デバイスにおいて、受信チケット (PRT) の正当性と利用者チェック処理の結果、チケットおよび利用者の正当なことが確認できなかった場合 (S414でNo) は、パーティション登録チケット (PRT) 受理エラーをパーティション作成・削除装置に通知 (S418) する。チケットおよび利用者の正当なことが確認できた場合 (S414でYes) は、受信したパーティション登録チケット (PRT) に記述されたルールに従いデバイス内のメモリ部におけるパーティション

の生成、または削除処理を実行する。この処理の詳細についても別フローを用いて後段で詳述する。

【0356】パーティション登録チケット (PRT) の記述に従って、パーティションの生成または削除処理に成功 (S416でYes) すると、PRT受理成功をパーティション作成・削除装置に通知 (S417) する。一方、パーティションの生成または削除処理に失敗 (S416でNo) した場合は、PRT受理エラーをパーティション作成・削除装置に通知 (S418) する。

【0357】パーティション作成・削除装置は、PRT受理結果を受信 (S404) し、PRT処理結果を判定し、PRT受理結果がエラーである場合 (S405でNo) は、エラーとして処理を終了し、PRT受理結果が成功 (S405でYes) であり、処理がパーティション生成である場合はパーティション初期データの書き込み処理 (S406, S419) を実行する。初期データの書き込み処理については、後段で別フローを用いて詳述する。パーティション初期データの書き込み処理が終了した場合、およびPRT受理結果が成功 (S405でYes) であり、処理がパーティション削除である場合はセッションクリアコマンドの送受信 (S407, S420) を実行し、デバイス側に生成した認証テーブルを破棄 (S421) し、処理を終了する。認証テーブルは、ステップS401, S410の相互認証処理において生成されるテーブルであり、その詳細は後述する。

【0358】このようにパーティション登録チケット (PRT) を利用して、デバイス内に新たなパーティションの生成、または生成済みのパーティションの削除が実行される。以下、当処理に含まれる相互認証処理 (S401, S410)、チケットの正当性と利用者チェック (S413)、パーティションの生成、削除処理 (S415)、パーティション初期データ書き込み処理 (S406, S419) の各処理について詳述する。

【0359】(相互認証処理) 図47のステップS401, S410において実行される相互認証処理について説明する。なお、以下に説明する相互認証処理は、他のチケット、すなわちファイル登録チケット (FRT: File Registration Ticket)、サービス許可チケット (SPT: Service Permission Ticket)、データアップデートチケット (DUT: Data Update Ticket) を使用したデバイスアクセス処理においても適宜必要に応じて行われる処理である。

【0360】相互認証処理の処理フローを図48に示す。図48において、左側がパーティションマネージャの認証装置、右側がデバイス (図5参照) の処理を示す。なお、パーティションマネージャの認証装置は、デバイスに対するデータ読み取り書き込み処理可能な装置 (ex. デバイスアクセス機器としてのリーダライタ、PC) であり、図10のリーダライタに相当する構成を有する。

【0361】図48のステップS431において、認証装置は、パーティション登録チケット（PRT）の利用に必要な認証方式をチケットから読み出して決定する。なお、認証態様は、チケット記載の認証方式に限らず、例えばアクセス機器（ex. リーダライタ）からの指定された方式に従ってデバイス認証、パーティション認証を決定してもよい。

【0362】決定した認証方式をA（1）～A（n）とする。パーティション登録チケット（PRT）を適用したパーティションの設定登録、あるいは削除処理においては様々な認証処理態様が設定される。例えばパーティションの設定登録についてはデバイスを対象とするデバイス認証を必要とし、パーティション削除の場合にはデバイス認証と削除対象となるパーティション認証の双方を必要とするなどである。このように処理に応じていずれか一方の認証、または両者の認証を必要とする設定をパーティション登録チケット（PRT）に記述することができる。PRT利用処理において必要とする認証方式については、パーティション登録チケット（PRT）の[Authentication Type]に記述され、認証装置は、パーティション登録チケット（PRT）の利用に必要な認証方式をチケットから読み出して決定し、認証処理手順をA（i）：A（1）～A（n）として定義する。

【0363】ステップS432において、最初の認証処理方式A（1）を読み出し、A（1）の認証方式がデバイス認証、パーティション認証であるかについて判別（S433）し、デバイス認証であればデバイス認証を実行（S434、S441）し、パーティション認証であればパーティション認証を実行（S435、S442）する。デバイスとの認証処理の結果、認証が成功しない場合は、エラーとして処理を終了する。認証が成功した場合は、ステップS437においてiをインクリメントしてステップS433に戻り、次の認証方式を判別して、方式に従って認証を実行する。これらの処理をA（1）～A（n）まで実行して、全ての認証が成功した場合に次のステップに進む。

【0364】デバイス認証処理について図49のフローに従って説明する。図49において、左側がパーティションマネージャのデバイス認証装置、右側がデバイス（図5参照）の処理を示す。なお、パーティションマネージャのデバイス認証装置は、デバイスに対するデータ読み取り書き込み処理可能な装置（ex. デバイスアクセス機器としてのリーダーライタ、PC）であり、図10のデバイスアクセス機器としてのリーダーライタに相当する構成を有する。

【0365】デバイス認証装置は、ステップS451において、パーティション登録チケット（PRT）に基づいてデバイス認証処理に公開鍵を用いた公開鍵認証方式を適用するか否かを判定する。デバイス認証は、公開鍵方式または共通鍵方式のいずれかにおいて実行され、チ

ケットにその実行方式についての指定が記述されている。

【0366】共通鍵方式の指定がある場合は、図49のステップS452～S455、S461～S465の処理は実行されず、ステップS456に進む。公開鍵方式の指定である場合は、デバイス認証装置は、ステップS452において公開鍵デバイス認証開始コマンドをデバイスに送信する。デバイスは、コマンドを受信（S461）すると、デバイスのメモリ部の公開鍵系デバイス鍵定義ブロック（図16参照）を参照して、デバイス対応公開鍵証明書（CERT DEV）が格納されているか否かを検証（S462）する。デバイス対応公開鍵証明書（CERT DEV）が格納されていない場合は、公開鍵方式の相互認証は実行できず、エラーと判定され処理は終了される。

【0367】デバイス対応公開鍵証明書（CERT DEV）が格納されていることが確認されると、ステップS453、S463において、デバイスマネージャ対応認証局（CA（DEV））の発行した公開鍵証明書を用いた相互認証および鍵共有処理が実行される。

【0368】公開鍵方式による相互認証および鍵共有処理について、図50を用いて説明する。図50は、公開鍵暗号方式である160ビット長の楕円曲線暗号（ECC）を用いた相互認証シーケンスを示している。図50では、公開鍵暗号方式としてECCを用いているが、公開鍵暗号方式の他方式を適用することも可能である。また、鍵サイズも160ビットでなくてもよい。図50において、まずBが、64ビットの乱数Rbを生成し、Aに送信する。これを受信したAは、新たに64ビットの乱数Raおよび標数pより小さい乱数Akを生成する。そして、ベースポイントGをAk倍した点Av=Ak×Gを求め、Ra、Rb、Av（X座標とY座標）に対する電子署名A. Sigを生成し、Aの公開鍵証明書とともにBに返送する。ここで、RaおよびRbはそれぞれ64ビット、AvのX座標とY座標がそれぞれ160ビットであるので、合計448ビットに対する電子署名を生成する。

【0369】公開鍵証明書を利用する際には、利用者は自己が保持する公開鍵証明書発行局（CA）の公開鍵を用い、当該公開鍵証明書の電子署名を検証し、電子署名の検証に成功した後に公開鍵証明書から公開鍵を取り出し、当該公開鍵を利用する。従って、公開鍵証明書を利用する全ての利用者は、共通の公開鍵証明書発行局（CA）の公開鍵を保持している必要がある。なお、電子署名の検証方法については、図13で説明したのでその詳細は省略する。

【0370】Aの公開鍵証明書、Ra、Rb、Av、電子署名A. Sigを受信したBは、Aが送信してきたRbが、Bが生成したものと一致するか検証する。その結果、一致していた場合には、Aの公開鍵証明書内の電子

署名を認証局の公開鍵で検証し、Aの公開鍵を取り出す。そして、取り出したAの公開鍵を用い電子署名A. Sigを検証する。電子署名の検証に成功した後、BはAを正当なものとして認証する。

【0371】次に、Bは、標数pより小さい乱数Bkを生成する。そして、ベースポイントGをBk倍した点Bv=Bk×Gを求め、Rb、Ra、Bv（X座標とY座標）に対する電子署名B. Sigを生成し、Bの公開鍵証明書とともにAに返送する。

【0372】Bの公開鍵証明書、Rb、Ra、Bv、電子署名B. Sigを受信したAは、Bが送信してきたRaが、Aが生成したものと一致するか検証する。その結果、一致していた場合には、Bの公開鍵証明書内の電子署名を認証局の公開鍵で検証し、Bの公開鍵を取り出す。そして、取り出したBの公開鍵を用い電子署名B. Sigを検証する。電子署名の検証に成功した後、AはBを正当なものとして認証する。

【0373】両者が認証に成功した場合には、BはBk×Av（Bkは乱数だが、Avは楕円曲線上の点であるため、楕円曲線上の点のスカラー倍計算が必要）を計算し、AはAk×Bvを計算し、これら点のX座標の下位64ビットをセッション鍵として以降の通信に使用する（共通鍵暗号を64ビット鍵長の共通鍵暗号とした場合）。もちろん、Y座標からセッション鍵を生成してもよいし、下位64ビットでなくてもよい。なお、相互認証後の秘密通信においては、送信データはセッション鍵で暗号化されるだけでなく、電子署名も付されることがある。

【0374】電子署名の検証や受信データの検証の際に、不正、不一致が見つかった場合には、相互認証が失敗したものとして処理を中断する。

【0375】このような相互認証処理において、生成したセッション鍵を用いて、送信データを暗号化して、相互にデータ通信を実行する。

【0376】図49に戻りフローの説明を続ける。ステップS453、S463において上述のような相互認証、鍵共有処理に成功すると、デバイスは、ステップS464において、デバイスのメモリ部のデバイス鍵領域（図18参照）に格納されたCRL\_DEV:排除デバイス(Device)、排除機器（デバイスアクセス機器としてのリーダライタ、PC等のチケットユーザ、チケット発行手段）の公開鍵証明書識別子（ex. シリアルナンバー:SN）を登録したリボケーションリスト（Revocation List (Certificate)）を参照して、通信相手であるデバイス認証装置がリボークされていないかを検証する。リボークされている場合は、パーティションの生成処理を許可できないので、エラーとして処理を終了する。

【0377】リボークされていない場合は、ステップS465において、相互認証および鍵共有処理において生成したセッション鍵Ksesと、通信相手（デバイス認

証装置を構成するデバイスアクセス機器としてのリーダライタ、PCなど）の公開鍵証明書中の識別名（DN:Distinguished Name）、シリアルナンバー、カテゴリをデバイスマネージャコード（DMC）をキーとして対応付けた認証テーブルに保存する。

【0378】一方、デバイス認証装置も、ステップS454において、デバイスがリボークされていないかをCRL\_DEV:排除デバイス(Device)、排除機器（デバイスアクセス機器としてのリーダライタ、PC等のチケットユーザ、チケット発行手段）の公開鍵証明書識別子（ex. シリアルナンバー:SN）を登録したリボケーションリスト（Revocation List (Certificate)）を参照して判定する。デバイス認証装置は、リボケーションリスト（CRL\_DEV）を登録局（RA(PAR)）から取得可能である。リボークされている場合は、パーティションの生成処理を許可できないので、エラーとして処理を終了する。

【0379】リボークされていない場合は、ステップS455において、相互認証および鍵共有処理において生成したセッション鍵Ksesと、通信相手（デバイス）の公開鍵証明書中の識別名（DN:Distinguished Name）、シリアルナンバー、カテゴリをデバイスマネージャコード（DMC）をキーとして対応付けた認証テーブルに保存する。

【0380】図51にデバイス内に生成される認証テーブルの例を示し、図52に認証装置としてのデバイスアクセス機器としてのリーダライタ（PCも可）に生成される認証テーブルの例を示す。

【0381】図51は、デバイス認証、および後段で説明するパーティション認証としてのパーティション1、2の認証が終了した時点のデバイス内に生成される認証テーブルの例である。グループ(Group)として、デバイス認証の場合はデバイスマネージャコード(DMC)、パーティション認証の場合はパーティションマネージャコード(PMC)が記録され、実行された各認証方式によってそれぞれのデータが格納される。公開鍵認証方式の場合は、前述したようにセッション鍵Ksesと、通信相手（デバイスアクセス機器としてのリーダライタ）の公開鍵証明書中の識別名(DN:Distinguished Name)、シリアルナンバー、カテゴリがテーブルに格納され、共通鍵認証の場合は、セッション鍵Ksesと、通信相手（デバイスアクセス機器としてのリーダライタ）の識別子(IDRW)が格納される。

【0382】認証テーブルはセッションのクリア時点で破棄される。デバイスはテーブルの情報を参照することによってデバイスおよび各パーティションの認証状態を確認することができ、使用すべきセッション鍵の確認が可能となる。

【0383】一方、図52は、デバイスアクセス機器としてのリーダライタ側の認証テーブルを示している。こ

の例もデバイス認証、およびパーティション認証としてのパーティション1, 2の認証が終了した時点の認証テーブルの例である。基本的構成は、デバイス内の認証テーブルと同様であり、グループ (Group) として、デバイス認証の場合はデバイスマネージャコード (DMC)、パーティション認証の場合はパーティションマネージャコード (PMC) が記録され、実行された各認証方式によってそれぞれのデータが格納される。公開鍵認証方式の場合は、前述したようにセッション鍵  $K_{ses}$  と、通信相手 (デバイス) の公開鍵証明書中の識別名 (DN: Distinguished Name)、シリアルナンバー、カテゴリーがテーブルに格納され、共通鍵認証の場合は、セッション鍵  $K_{ses}$  と、通信相手 (デバイス) の識別子 (ID<sub>RW</sub>) が格納される。リーダライタ側の認証テーブルもセッションのクリア時点で破棄される。デバイスアクセス機器としてのリーダライタ側においても、認証テーブルの情報を参照することによってデバイスおよびパーティションの認証状態の有無を判定可能となり、使用すべきセッション鍵の確認が可能となる。

【0384】図49に戻り、デバイス認証処理の説明を続ける。デバイス認証装置はステップS451において、デバイス認証方式が公開鍵方式でないと判定されると、デバイス認証装置はステップS456において、共通鍵デバイス認証コマンドをデバイスに出力する。デバイスは、コマンドを受信 (S466) すると、デバイスのメモリ部の共通鍵系デバイス鍵定義ブロック (図16参照) を参照して共通鍵認証に使用する双方向個別鍵認証用マスター鍵 (MK<sub>auth\_DEV</sub>) が格納されているか否かを検証 (S467) する。双方向個別鍵認証用マスター鍵 (MK<sub>auth\_DEV</sub>) が格納されていない場合は、共通鍵方式の相互認証は実行できず、エラーと判定され処理は終了される。

【0385】双方向個別鍵認証用マスター鍵 (MK<sub>auth\_DEV</sub>) が格納されていることが確認されると、ステップS457、S468において、マスター鍵を使用した相互認証および鍵共有処理が実行される。

【0386】マスター鍵を使用した共通鍵方式による相互認証および鍵共有処理について、図53を用いて説明する。図53では、AおよびBがマスター鍵を使用した共通鍵方式の認証を実行するエンティティであり、Aは、自己の識別子ID<sub>A</sub>、双方向個別鍵認証用共通鍵K<sub>a</sub>、双方向個別鍵認証用マスター鍵MK<sub>b</sub>を有し、Bは、自己の識別子ID<sub>B</sub>、双方向個別鍵認証用共通鍵K<sub>b</sub>、双方向個別鍵認証用マスター鍵MK<sub>a</sub>を有する。なお、図53の例では、共通鍵暗号方式としてDESアルゴリズム (ex. DES, トリプルDES) を用いているが、同様な共通鍵暗号方式であれば他の暗号方式の適用も可能である。

【0387】まず、Bが64ビットの乱数R<sub>b</sub>を生成し、R<sub>b</sub>および自己のIDであるID<sub>B</sub>をAに送信す

る。これを受信したAは、新たに64ビットの乱数R<sub>a</sub>を生成し、双方向個別鍵認証用マスター鍵MK<sub>b</sub>によるID<sub>B</sub>のDES暗号化処理により双方向個別鍵認証用共通鍵K<sub>b</sub>を取得する。さらに、鍵K<sub>a</sub>、K<sub>b</sub>を用いて、R<sub>a</sub>、R<sub>b</sub>、ID<sub>A</sub>、ID<sub>B</sub>の順に、DESのCBCモードでデータを暗号化し、自己の識別子ID<sub>A</sub>とともにBに返送する。

【0388】これを受信したBは、まず、双方向個別鍵認証用マスター鍵MK<sub>a</sub>によるID<sub>A</sub>のDES暗号化処理により双方向個別鍵認証用共通鍵K<sub>a</sub>を取得する。さらに、受信データを鍵K<sub>a</sub>、K<sub>b</sub>で復号する。復号して得られたR<sub>a</sub>、R<sub>b</sub>、ID<sub>A</sub>、ID<sub>B</sub>の内、R<sub>b</sub>およびID<sub>B</sub>が、Bが送信したものと一致するか検証する。この検証に通った場合、BはAを正当なものとして認証する。

【0389】次にBは、セッション鍵として使用する64ビットの乱数K<sub>ses</sub>を生成し、鍵K<sub>b</sub>、K<sub>a</sub>を用いて、R<sub>b</sub>、R<sub>a</sub>、ID<sub>B</sub>、ID<sub>A</sub>、K<sub>ses</sub>の順に、DESのCBCモードでデータを暗号化し、Aに返送する。

【0390】これを受信したAは、受信データを鍵K<sub>a</sub>、K<sub>b</sub>で復号する。復号して得られたR<sub>a</sub>、R<sub>b</sub>、ID<sub>A</sub>、ID<sub>B</sub>がAが送信したものと一致するか検証する。この検証に通った場合、AはBを正当なものとして認証する。互いに相手を認証した後は、セッション鍵K<sub>ses</sub>は、認証後の秘密通信のための共通鍵として利用される。

【0391】なお、受信データの検証の際に、不正、不一致が見つかった場合には、相互認証が失敗したものとして処理を中止する。

【0392】本発明のシステムの格納データに対応付けたマスター鍵を使用した共通鍵認証について、データの流れを説明する図を図54に示す。図54に示すように、デバイスアクセス機器としてのリーダライタ (R/W) が64ビットの乱数R<sub>b</sub>を生成し、R<sub>b</sub>および自己のIDであるID<sub>rw</sub>をデバイス (Device) に送信する。これを受信したデバイス (Device) は、新たに64ビットの乱数R<sub>a</sub>を生成し、双方向個別鍵認証用マスター鍵MK<sub>auth\_DEV\_A</sub>によるID<sub>rw</sub>のDES暗号化処理により双方向個別鍵認証用共通鍵K<sub>auth\_DEV\_A</sub>を取得する。さらに、鍵K<sub>auth\_DEV\_A</sub>、K<sub>auth\_DEV\_B</sub>を用いて、R<sub>a</sub>、R<sub>b</sub>、ID<sub>rw</sub>の順に、暗号アルゴリズムとして、例えばDES-CBCモードでデータを暗号化し、自己の識別子ID<sub>m</sub>とともにデバイスアクセス機器としてのリーダライタ (R/W) に返送する。

【0393】これを受信したリーダライタ (R/W) は、まず、双方向個別鍵認証用マスター鍵MK<sub>auth\_DEV\_B</sub>によるID<sub>m</sub>のDES暗号化処理により双方向個別鍵認証用共通鍵K<sub>auth\_DEV\_B</sub>を取得する。さらに、受信データを鍵K<sub>auth\_DEV\_A</sub>、K<sub>auth\_DEV\_B</sub>で復号する。復号



して得られたRbおよびIDrwが、デバイスアクセス機器としてのリーダライタ(R/W)が送信したものと一致するか検証する。この検証に通った場合、リーダライタ(R/W)はデバイス(Device)を正当なものとして認証する。

【0394】次にリーダライタ(R/W)は、セッション鍵として使用する64ビットの乱数Ksesを生成し、双方向個別鍵認証用共通鍵Kauth\_DEV\_A、Kauth\_DEV\_Bを用いて、Rb、Ra、Ksesの順に、DESアルゴリズムとしての例えばトリプルDESモードでデータを暗号化し、デバイス(Device)に返送する。

【0395】これを受信したデバイスは、受信データを双方向個別鍵認証用共通鍵Kauth\_DEV\_A、Kauth\_DEV\_Bで復号する。復号して得られたRa、Rb、IDrwがデバイス(Device)が送信したものと一致するか検証する。この検証に通った場合、デバイス(Device)はリーダライタ(R/W)を正当なものとして認証し、認証後、セッション鍵Ksesを秘密通信のための共通鍵として利用する。

【0396】なお、デバイスの固有値としてのIDmは、先に図6のデバイスメモリフォーマットを使用して説明したようにデバイスマネージャ管理領域の格納データに基づく値を適用することができる。

【0397】上述したように、マスター鍵を使用した共通鍵方式による相互認証および鍵共有処理によれば、通信相手方のマスター鍵に基づく処理を実行して生成した通信相手の個別鍵と、自己の個別鍵の2つの鍵を共通鍵として設定し、設定した2つの鍵を用いて共通鍵方式による相互認証を実行する構成であるので、デバイスまたはアクセス装置に予め共通鍵が格納された従来の共通鍵認証構成に比較してよりセキュアな認証システムおよび方法が実現される。

【0398】図49に戻りフローの説明を続ける。ステップS457、S468において上述のような相互認証、鍵共有処理に成功すると、デバイスは、ステップS469において、デバイスのメモリ部のデバイス鍵領域(図18参照)に格納されたIRL\_DEV:排除デバイス(Device)、排除機器(デバイスアクセス機器としてのリーダライタ、PC等のチケットユーザ、チケット発行手段)の識別子(ID)を登録したリボケーションリスト(Revocation List(ID))を参照して、通信相手であるデバイス認証装置がリボークされていないかを検証する。リボークされている場合は、パーティションの生成処理を許可できないので、エラーとして処理を終了する。

【0399】リボークされていない場合は、ステップS470において、相互認証および鍵共有処理において生成したセッション鍵Ksesと、通信相手(デバイス認証装置を構成するデバイスアクセス機器としてのリーダ

ライタ、PCなど)の識別情報(IDrw)をデバイスマネージャコード(DMC)をキーとして対応付けた認証テーブル(図51参照)に保存する。

【0400】一方、デバイス認証装置も、ステップS458において、デバイスがリボークされていないかをIRL\_DEV:排除デバイス(Device)、排除機器(デバイスアクセス機器としてのリーダライタ、PC等のチケットユーザ、チケット発行手段)の識別子(ID)を登録したリボケーションリスト(Revocation List(ID))を参照して判定する。デバイス認証装置は、リボケーションリスト(IRL\_DEV)を登録局(RA(PAR))から取得可能である。リボークされている場合は、パーティションの生成処理を許可できないので、エラーとして処理を終了する。

【0401】リボークされていない場合は、ステップS459において、相互認証および鍵共有処理において生成したセッション鍵Ksesと、通信相手(デバイス)の識別情報(IDm)をデバイスマネージャコード(DMC)をキーとして対応付けた認証テーブル(図52参照)に保存する。

【0402】以上の処理が、パーティションマネージャの管轄するデバイスアクセス機器としてのリーダライタとデバイス間において実行されるデバイス認証処理である。

【0403】次に、図48のステップS435、S442において実行されるパーティション認証処理について、図55、図56を用いて説明する。パーティション登録チケットを適用したパーティションの設定登録、または削除処理においては、先に説明したように処理に応じてデバイス認証、パーティション認証のいずれか、または両者の認証を必要とする。これらの設定はパーティション登録チケット(PRT)に記述される。パーティション登録チケット(PRT)にパーティション認証を実行する記述がある場合は、パーティション認証が実行される。

【0404】図55、図56の処理フローの各ステップについて説明する。図55において、左側がパーティションマネージャのパーティション認証装置、右側がデバイス(図5参照)の処理を示す。なお、パーティションマネージャのパーティション認証装置は、デバイスに対するデータ読み取り書き込み処理可能な装置(ex. デバイスアクセス機器としてのリーダライタ、PC)であり、図10のデバイスアクセス機器としてのリーダライタに相当する構成を有する。

【0405】パーティション認証装置は、ステップS471において、デバイスに対して認証対象となるパーティションAの存在確認を実行するパーティションA存在チェックコマンドを出力する。コマンドを受領(S481)したデバイスは、デバイスのメモリ部内にパーティションAが存在するか否かをチェック(S482)す

る。ここでパーティションの識別子Aとしては例えばパーティションマネージャコード (PMC) が使用され、デバイスは、例えばパーティション定義ブロック (PDB : Partition Definition Block) の格納PMCに基づいてパーティションの有無を判定することができる。デバイスによるパーティションの有無が判定されると、チェック結果がパーティション認証装置に送信される。

【0406】チェック結果を受領 (S472) したパーティション認証装置は、チェック結果を検証 (S473) し、パーティションの存在しないとの結果を受領した場合は、認証不可であるので、エラー終了する。チェック結果がパーティションが存在することを示した場合は、パーティション認証装置は、ステップS474において、パーティション登録チケット (PRT) に基づいてパーティション認証処理に公開鍵を用いた公開鍵認証方式を適用するか否かを判定する。パーティション認証は、前述のデバイス認証と同様、公開鍵方式または共通鍵方式のいずれかにおいて実行され、チケットにその実行方式についての指定が記述されている。

【0407】共通鍵方式の指定がある場合は、図55のステップS475～S478、S484～S488の処理は実行されず、ステップS491に進む。公開鍵方式の指定である場合は、パーティション認証装置は、ステップS475において公開鍵パーティションA認証開始コマンドをデバイスに送信する。デバイスは、コマンドを受信 (S484) すると、デバイスのメモリ部の公開鍵系パーティション鍵定義ブロック (図21参照) を参照して、パーティションA対応公開鍵証明書 (CERT PAR) が格納されているか否かを検証 (S485) する。パーティションA対応公開鍵証明書 (CERT PAR) が格納されていない場合は、公開鍵方式の相互認証は実行できず、エラーと判定され処理は終了される。

【0408】パーティションA対応公開鍵証明書 (CERT PAR) が格納されていることが確認されると、ステップS476、S486において、パーティションマネージャ対応認証局 (CA (PAR)) の発行した公開鍵証明書を用いた相互認証および鍵共有処理が実行される。

【0409】公開鍵方式による相互認証および鍵共有処理については、先のデバイス認証処理において説明した図50に示すシーケンスと同様であるので説明を省略する。ただし、パーティション認証において利用する公開鍵証明書は、パーティションマネージャ対応認証局 (CA (PAR)) の発行した公開鍵証明書であり、この公開鍵証明書の署名検証には、パーティションマネージャ対応認証局 (CA (PAR)) の公開鍵 (PUB CA (PAR)) を用いる。公開鍵 (PUB CA (PAR)) は、パーティション鍵領域 (図23参照) に格納されている。このような相互認証処理において、生成し

たセッション鍵を用いて、送信データを暗号化して、相互にデータ通信を実行する。

【0410】ステップS476、S486において図50に示すシーケンスに従った相互認証、鍵共有処理に成功すると、デバイスは、ステップS487において、デバイスのメモリ部のパーティション鍵領域 (図23参照) に格納されたCRL\_PAR : 排除デバイス (Device)、排除機器 (デバイスアクセス機器としてのリーダライタ、PC等のチケットユーザ、チケット発行手段) の公開鍵証明書識別子 (ex. シリアルナンバー : SN) を登録したリボケーションリスト (Revocation List (Certificate)) を参照して、通信相手であるパーティション認証装置がリボークされていないかを検証する。リボークされている場合は、パーティションの生成処理あるいは削除処理を許可できないので、エラーとして処理を終了する。

【0411】リボークされていない場合は、ステップS488において、相互認証および鍵共有処理において生成したセッション鍵Ksesと、通信相手 (パーティション認証装置を構成するデバイスアクセス機器としてのリーダライタ、PCなど) の公開鍵証明書の識別名

(DN : Distinguished Name)、シリアルナンバー、カテゴリーをパーティションマネージャコード (PMC) をキーとして対応付けた認証テーブルに保存する。

【0412】一方、パーティション認証装置も、ステップS477において、デバイスがリボークされていないかをCRL\_PAR : 排除デバイス (Device)、排除機器 (デバイスアクセス機器としてのリーダライタ、PC等のチケットユーザ、チケット発行手段) の公開鍵証明書識別子 (ex. シリアルナンバー : SN) を登録したリボケーションリスト (Revocation List (Certificate)) を参照して判定する。デバイス認証装置は、リボケーションリスト (CRL\_PAR) を登録局 (RA (PAR)) から取得可能である。リボークされている場合は、パーティションの生成処理あるいは削除処理を許可できないので、エラーとして処理を終了する。

【0413】リボークされていない場合は、ステップS478において、相互認証および鍵共有処理において生成したセッション鍵Ksesと、通信相手 (デバイス) の公開鍵証明書の識別名 (DN : Distinguished Name)、シリアルナンバー、カテゴリーをパーティションマネージャコード (PMC) をキーとして対応付けた認証テーブルに保存する。この結果、例えば図51に示す認証テーブルがデバイス内に生成され、図52に示す認証テーブルがパーティション認証装置としてのデバイスアクセス機器としてのリーダライタ (PCも可) に生成される。図51、図52は、デバイス認証、およびパーティション認証としてのパーティション1、2の認証が終了した時点のデバイス内およびデバイスアクセス機器としてのリーダライタ内に生成される認証テーブルの例

である。

【0414】パーティション認証の場合はパーティションマネージャコード (PMC) が記録され、実行された各認証方式によってそれぞれのデータが格納される。公開鍵認証方式の場合は、前述したようにセッション鍵  $K_{ses}$  と、通信相手の公開鍵証明書中の識別名 (DN: Distinguished Name)、シリアルナンバー、カテゴリーがテーブルに格納され、共通鍵認証の場合は、セッション鍵  $K_{ses}$  と、通信相手の識別子が格納される。認証テーブルはセッションのクリア時点で破棄される。デバイスおよびデバイスアクセス機器としてのリーダライタはテーブルの情報を参照することによってデバイスおよび各パーティションの認証状態を確認することができ、使用すべきセッション鍵の確認が可能となる。

【0415】図55、図56のフローを用いて、パーティション認証処理の説明を続ける。パーティション認証装置はステップS474において、パーティション認証方式が公開鍵方式でないと判定されると、パーティション認証装置はステップS491において、共通鍵パーティションA認証コマンドをデバイスに出力する。デバイスは、コマンドを受信 (S501) すると、デバイスのメモリ部の共通鍵系パーティション鍵定義ブロック (図22参照) を参照して共通鍵認証に使用する双方向個別鍵認証用マスター鍵 ( $MK_{auth\_PAR\_A}$ ) が格納されているか否かを検証 (S502) する。双方向個別鍵認証用マスター鍵 ( $MK_{auth\_PAR\_A}$ ) が格納されていない場合は、共通鍵方式の相互認証は実行できず、エラーと判定され処理は終了される。

【0416】双方向個別鍵認証用マスター鍵 ( $MK_{auth\_PAR\_A}$ ) が格納されていることが確認されると、ステップS492、S503において、マスター鍵を使用した相互認証および鍵共有処理が実行される。共通鍵方式による相互認証および鍵共有処理については、先のデバイス認証において、図53、図54を用いて説明したシーケンスと同様であるので、説明を省略する。ただし、パーティション認証の場合に適用する鍵は、パーティション鍵定義ブロック (図22参照) に定義され、パーティション鍵領域 (図23参照) に格納された双方向個別鍵認証用共通鍵 ( $K_{auth\_PAR\_B}$ )、および双方向個別鍵認証用マスター鍵 ( $MK_{auth\_PAR\_A}$ ) である。

【0417】ステップS492、S503において共通鍵方式の相互認証、鍵共有処理に成功すると、デバイスは、ステップS504において、デバイスのメモリ部のパーティション鍵領域 (図23参照) に格納された  $IRL\_PAR$ : 排除デバイス (Device)、排除機器 (デバイスアクセス機器としてのリーダライタ、PC等のチケットユーザ、チケット発行手段) の識別子 (ID) を登録したリボケーションリスト (Revocation List (ID)) を参照して、通信相手であるパーティション認証装置がリボークされていないかを検証する。リボークされている場合

は、パーティションの生成処理または削除処理を許可できないので、エラーとして処理を終了する。

【0418】リボークされていない場合は、ステップS505において、相互認証および鍵共有処理において生成したセッション鍵  $K_{ses}$  と、通信相手 (デバイス認証装置を構成するデバイスアクセス機器としてのリーダライタ、PCなど) の識別情報 ( $ID_{rw}$ ) をパーティションマネージャコード (PMC) をキーとして対応付けた認証テーブル (図51参照) に保存する。

【0419】一方、パーティション認証装置も、ステップS493において、デバイスがリボークされていないかを  $IRL\_PAR$ : 排除デバイス (Device)、排除機器 (デバイスアクセス機器としてのリーダライタ、PC等のチケットユーザ、チケット発行手段) の識別子 (ID) を登録したリボケーションリスト (Revocation List (ID)) を参照して判定する。パーティション認証装置は、リボケーションリスト ( $IRL\_PAR$ ) を登録局 ( $RA(PAR)$ ) から取得可能である。リボークされている場合は、パーティションの生成処理または削除処理を許可できないので、エラーとして処理を終了する。

【0420】リボークされていない場合は、ステップS494において、相互認証および鍵共有処理において生成したセッション鍵  $K_{ses}$  と、通信相手 (デバイス) の識別情報 ( $ID_m$ ) をパーティションマネージャコード ( $DMC$ ) をキーとして対応付けた認証テーブル (図52参照) に保存する。

【0421】以上の処理が、パーティションマネージャの管轄するデバイスアクセス機器としてのリーダライタとデバイス間において実行されるパーティション認証処理である。このような相互認証により、デバイスまたはパーティションとデバイスアクセス機器としてのリーダライタ間の認証が成立し、セッション鍵の共有が達成され、通信データのセッション鍵による暗号化通信が可能となる。

【0422】なお、上述したデバイス認証処理、パーティション認証処理は、他のチケット、すなわちファイル登録チケット ( $FRT$ : File Registration Ticket)、サービス許可チケット ( $SPT$ : Service Permission Ticket)、データアップデートチケット ( $DUT$ : Data Update Ticket) を使用したデバイスアクセスを実行する際にも適宜必要に応じて行われる処理である。これらについては、後段の各チケットを利用した処理の説明中で述べる。

【0423】(チケットの正当性と利用者チェック) 次に、図47のパーティションの作成、削除処理フロー中のステップS413のデバイスにおけるチケットの正当性と利用者チェック処理の詳細について図57、図58のフローを用いて説明する。

【0424】なお、以下に説明するチケットの正当性と利用者チェック処理は、他のチケット、すなわちファイ

ル登録チケット (FRT: File Registration Ticket)、サービス許可チケット (SPT: Service Permission Ticket)、データアップデートチケット (DUT: Data Update Ticket) を使用したデバイスアクセス処理においても適宜必要に応じて行われる処理であり、図57、図58のフローは、各チケットに共通の処理フローとして構成してある。

【0425】チケットの正当性と利用者チェック処理は、デバイスとの通信を実行しているチケットユーザ (ex. デバイスアクセス機器としてのリーダライタ、PC等) から受信したチケットに基づいてデバイス (図5参照) が実行する処理である。デバイスは、チケットの正当性と利用者チェック処理においてチケットおよびチケットユーザ (ex. デバイスアクセス機器としてのリーダライタ、PC等) である利用者の正当性を確認した後、チケットに記述された制限範囲内の処理を許可する。

【0426】図57、図58のフローを用いてチケットの正当性と利用者チェック処理の詳細について説明する。チケットをチケットユーザ (ex. デバイスアクセス機器としてのリーダライタ、PC等) から受信したデバイスは、図57のステップS511において、チケットタイプを検証しチケットがパーティション登録チケット (PRT: Partition Registration Ticket) であるか否かを判定する。チケットタイプは、各チケットに記録されている (図26、図27、図28、図31、図32参照)。

【0427】チケットタイプがパーティション登録チケット (PRT: Partition Registration Ticket) である場合は、ステップS512～S514を実行し、パーティション登録チケット (PRT: Partition Registration Ticket) でない場合は、ステップS515に進む。

【0428】チケットタイプがパーティション登録チケット (PRT: Partition Registration Ticket) である場合は、ステップS512において、チケットに記述されたIntegrity Check Type (チケット (Ticket) の正当性検証値の種別 (公開鍵方式 (Public) / 共通鍵方式 (Common))) の設定が公開鍵方式 (Public) であるか否かを判定する。

【0429】正当性検証値の種別 (Integrity Check Type) が公開鍵方式 (Public) である場合、ステップS513に進み、各種処理を実行する。ステップS513で実行する処理は、まず、デバイスマネージャ対応認証局 (CA (DEV)) の公開鍵 PUB CA (DEV) を用いたチケット発行者 (Ticket Issuer) の公開鍵証明書 (CERT) の検証処理である。

【0430】前述したように、パーティション登録チケット (PRT) 発行手段 (PRT Issuer) の発行したチケット (Ticket) を、チケットユーザに対して送信する際

には、公開鍵方式の場合、パーティション登録チケット (PRT) 発行手段 (PRT Issuer) の公開鍵証明書 (CERT\_PRTI) も一緒にデバイスに送信される。なお、PRT発行手段の公開鍵証明書 (CERT\_PRTI) の属性 (Attribute) は、パーティション登録チケット (PRT) 発行手段 (PRT User) の識別子 (PRTIC) と一致する。

【0431】公開鍵証明書 (図11参照) にはデバイスマネージャ対応認証局 (CA (DEV)) の秘密鍵で実行された署名が付加されており、この署名をデバイスマネージャ対応認証局 (CA (DEV)) の公開鍵 PUB CA (DEV) を用いて検証する。署名生成、検証は、例えば先に説明した図12、図13のフローに従った処理として実行される。この署名検証により、チケット発行者の公開鍵証明書 (CERT) が改竄されたものでない正当な公開鍵証明書 (CERT) であるか否かが判定される。

【0432】さらに、ステップS513では、署名検証により正当性が確認されたチケット発行手段の公開鍵証明書 (CERT) のオプション領域に記録されたユーザのカテゴリとしてのコードが、デバイス内のDKDB (Device Key Definition Block) (PUB) に記録されたチケット発行手段コード (PRTIC: PRT Issuer Code) と一致するか否かを判定する。

【0433】公開鍵証明書には、図11の公開鍵証明書の説明の欄で記述したように、各チケット (PRT, FRT, SPTなど) の発行手段であるチケット発行手段 (Ticket Issuer) の所属コード、この場合、PRTIC (PRT Issuer Code) が記録されている。このオプション領域のコードとデバイス内のDKDB (Device Key Definition Block) (PUB) に記録されたチケット発行手段コード (PRTIC: PRT Issuer Code) の一致を確認することで、受信チケット (PRT) が正当なチケット発行手段によって発行されたチケットであることを確認する。

【0434】さらに、デバイスは、デバイスのメモリ部内のデバイス鍵領域 (図18参照) に格納されたCRL\_DEV (排除デバイス (Device)、排除機器 (デバイスアクセス機器としてのリーダライタ、PC等のチケットユーザ、チケット発行手段) の公開鍵証明書識別子 (ex. シリアルナンバ: SN) を登録したリボケーションリスト (Revocation List (Certificate))) を参照して、チケット発行手段 (Ticket Issuer) がリボークされていないかを判定する。

【0435】さらに、受信チケットであるパーティション登録チケット (PRT) (図26参照) に記録された署名、すなわちIntegrity Check Value (チケット (Ticket) の正当性検証値 (公開鍵方式: 署名 (Signature))) の検証を実行し、チケットが改竄されていないかを確認する。署名検証は、先の公開鍵証明書の署名検証と同様、例えば図13のフローと同様のシーケンスに

従って実行される。

【0436】以上、(1) チケット発行者 (Ticket Issuer) の公開鍵証明書 (CERT) が改竄されたものでない正当な公開鍵証明書 (CERT) であること、

(2) チケット発行者 (Ticket Issuer) の公開鍵証明書 (CERT) のオプション領域に記録されたコードと、デバイス内のDKDB (Device Key Definition Block) (PUB)に記録されたチケット発行手段コード (PRTIC:PRT Issuer Code) の一致、(3) チケット発行手段 (Ticket Issuer) がリポーカされていないこと、(4) 受信チケット (PRT) の署名 (Signature) の検証によりチケットが改竄のないことの確認。以上のすべての確認がなされたことを条件としてチケットの正当性検証成功とする。上記(1)～(4)のいずれかが確認されない場合は、チケットの正当性の確認が得られないと判定され、パーティション登録チケット (PRT: Partition Registration Ticket) を利用した処理は中止される。

【0437】また、ステップS512において、チケットに記述されたIntegrity Check Type (チケット (Ticket) の正当性検証値の種別 (公開鍵方式(Public) /共通鍵方式(Common))) の設定が共通鍵方式(Common)であると判定された場合は、ステップS514に進みMAC (Message Authentication Code) 検証を実行する。デバイスは、デバイスのデバイス鍵領域 (図18参照) に格納されたパーティション登録チケット (PRT) のMAC検証用鍵: K<sub>p r t</sub>を使用してチケットのMAC検証処理を実行する。

【0438】図59にDES暗号処理構成を用いたMAC値生成例を示す。図59の構成に示すように対象となるメッセージを8バイト単位に分割し、(以下、分割されたメッセージをM1、M2、・・・、MNとする)、まず、初期値 (Initial Value (以下、IVとする)) とM1を排他的論理和する (その結果をI1とする)。次に、I1をDES暗号化部に入れ、MAC検証用鍵: K<sub>p r t</sub>を用いて暗号化する (出力をE1とする)。続けて、E1およびM2を排他的論理和し、その出力I2をDES暗号化部へ入れ、鍵K<sub>p r t</sub>を用いて暗号化する (出力E2)。以下、これを繰り返し、全てのメッセージに対して暗号化処理を施す。最後に出てきたENがメッセージ認証符号 (MAC (Message Authentication Code)) となる。なお、メッセージとしては、検証対象となるデータを構成する部分データが使用可能である。

【0439】改竄のないことが保証された例えばデータ送信側がデータ生成時に生成したICV (Integrity Check Value) と、データ受信側が受信データに基づいて生成したICVとを比較して同一のICVが得られればデータに改竄のないことが保証され、ICVが異なれば、改竄があったと判定される。改竄のないことが保証

された例えばデータ送信側がデータ生成時に生成したICVは、図26のパーティション登録チケット (PRT) のフォーマットに関する記述において説明したように、PRTのICV (Integrity Check Value) フィールドに格納されている。デバイスが生成したICVと受信チケット (PRT) に格納されたICVとを比較して一致していればチケットの正当性ありと判定し、不一致の場合はチケット改竄ありと判定し、チケットを利用した処理を中止する。

【0440】上述の処理によってチケットに記述されたIntegrity Check Typeが共通鍵方式である場合のチケット検証処理が完了する。

【0441】図57のフローに戻り、チケットの正当性と利用者チェック処理について説明を続ける。ステップS511において、チケットタイプがパーティション登録チケット (PRT: Partition Registration Ticket) でないと判定された場合は、ステップS515においてチケットタイプを検証しチケットがファイル登録チケット (FRT: File Registration Ticket) であるか否かを判定する。

【0442】チケットタイプがファイル登録チケット (FRT: File Registration Ticket) である場合は、ステップS516～S518を実行し、ファイル登録チケット (FRT: File Registration Ticket) でない場合は、ステップS519に進む。

【0443】チケットタイプがファイル登録チケット (FRT: File Registration Ticket) である場合は、ステップS516において、チケットに記述されたIntegrity Check Type (チケット (Ticket) の正当性検証値の種別 (公開鍵方式(Public) /共通鍵方式(Common))) の設定が公開鍵方式(Public)であるか否かを判定する。

【0444】正当性検証値の種別 (Integrity Check Type) が公開鍵方式(Public)である場合、ステップS517に進み、各種処理を実行する。ステップS517で実行する処理は、まず、パーティションマネージャ対応認証局 (CA (PAR)) の公開鍵PUB CA (PAR) を用いたチケット発行者 (Ticket Issuer) の公開鍵証明書 (CERT) の検証処理である。

【0445】ファイル登録チケット (FRT) 発行手段 (FRT Issuer) の発行したチケット (Ticket) を、チケットユーザに対して送信する際には、公開鍵方式の場合、ファイル登録チケット (FRT) 発行手段 (FRT Issuer) の公開鍵証明書 (CERT\_FRTI) も一緒にデバイスに送信される。なお、FRT発行手段の公開鍵証明書 (CERT\_FRTI) の属性 (Attribute) は、ファイル登録チケット (FRT) 発行手段 (FRT Issuer) の識別子 (FRTIC) と一致する。

【0446】公開鍵証明書 (図11参照) にはパーティションマネージャ対応認証局 (CA (PAR)) の秘密鍵で実行された署名が付加されており、この署名をパー

ティションマネージャ対応認証局 (CA (PAR)) の公開鍵 PUB CA (PAR) を用いて検証する。署名生成、検証は、例えば先に説明した図 12、図 13 のフローに従った処理として実行される。この署名検証により、チケット発行者の公開鍵証明書 (CERT) が改竄されたものでない正当な公開鍵証明書 (CERT) であるか否かが判定される。

【0447】さらに、ステップ S517 では、署名検証により正当性が確認されたチケット発行手段の公開鍵証明書 (CERT) のオプション領域に記録されたユーザの所属コードと、デバイス内の PKDB (Partition Key Definition Block) (PUB) に記録されたチケット発行手段コード (FRTIC : FRT Issuer Code) と一致するか否かを判定する。

【0448】公開鍵証明書には、図 11 の公開鍵証明書の説明の欄で記述したように、各チケット (PRT, FRT, SPT など) の発行手段であるチケット発行手段 (Ticket Issuer) の所属コード、この場合、FRTIC (FRT Issuer Code) が記録されている。このオプション領域のコードとデバイス内の PKDB (Partition Key Definition Block) (PUB) に記録されたチケット発行手段コード (FRTIC : FRT Issuer Code) の一致を確認することで、受信チケット (FRT) が正当なチケット発行手段によって発行されたチケットであることを確認する。

【0449】さらに、デバイスは、デバイスのメモリ部のパーティション鍵領域 (図 23 参照) に格納された CRL\_PAR (排除デバイス (Device)、排除機器 (デバイスアクセス機器としてのリーダライタ、PC 等のチケットユーザ、チケット発行手段) の公開鍵証明書識別子 (ex. シリアルナンバ : SN) を登録したリボケーションリスト (Revocation List (Certificate)) を参照して、チケット発行手段 (Ticket Issuer) がリボークされていないかを判定する。

【0450】さらに、受信チケットであるファイル登録チケット (FRT) (図 27 参照) に記録された署名、すなわち Integrity Check Value (チケット (Ticket) の正当性検証値 (公開鍵方式 : 署名 (Signature))) の検証を実行し、チケットが改竄されていないかを確認する。署名検証は、先の公開鍵証明書の署名検証と同様、例えば図 13 のフローと同様のシーケンスに従って実行される。

【0451】以上、(1) チケット発行者 (Ticket Issuer) の公開鍵証明書 (CERT) が改竄されたものでない正当な公開鍵証明書 (CERT) であること、

(2) チケット発行者 (Ticket Issuer) の公開鍵証明書 (CERT) のオプション領域に記録されたコードと、デバイス内の PKDB (Partition Key Definition Block) (PUB) に記録されたチケット発行手段コード (FRTIC : FRT Issuer Code) の一致、(3) チケ

ット発行手段 (Ticket Issuer) がリボークされていないこと、(4) 受信チケット (FRT) の署名 (Signature) の検証によりチケットが改竄のないことの確認。以上のすべての確認がなされたことを条件としてファイル登録チケット (FRT) の正当性検証成功とする。上記 (1) ~ (4) のいずれかが確認されない場合は、ファイル登録チケット (FRT) の正当性の確認が得られないと判定され、ファイル登録チケット (FRT) を利用した処理は中止される。

【0452】また、ステップ S516 において、チケットに記述された Integrity Check Type (チケット (Ticket) の正当性検証値の種別 (公開鍵方式 (Public) / 共通鍵方式 (Common))) の設定が共通鍵方式 (Common) であると判定された場合は、ステップ S518 に進み MAC (Message Authentication Code) 検証を実行する。デバイスは、デバイスのパーティション鍵領域 (図 23 参照) に格納されたファイル登録チケット (FRT) の MAC 検証用鍵 : K<sub>f r t</sub> を使用してチケットの MAC 検証処理を実行する。MAC 検証処理は、先に説明した図 59 の DES 暗号処理構成を用いた MAC 値生成処理に従って実行される。

【0453】改竄のないことが保証された例えばデータ送信側がデータ生成時に生成した ICV (Integrity Check Value) と、データ受信側が受信データに基づいて生成した ICV とを比較して同一の ICV が得られればデータに改竄のないことが保証され、ICV が異なれば、改竄があったと判定される。改竄のないことが保証された例えばデータ送信側がデータ生成時に生成した ICV は、図 27 のファイル登録チケット (FRT) のフォーマットに関する記述において説明したように、FRT の ICV (Integrity Check Value) フィールドに格納されている。デバイスが生成した ICV と受信チケット (FRT) に格納された ICV とを比較して一致していればチケットの正当性ありと判定し、不一致の場合はチケット改竄ありと判定し、チケットを利用した処理を中止する。

【0454】上述の処理によってチケットに記述された Integrity Check Type が共通鍵方式である場合のファイル登録チケット (FRT) 検証処理が完了する。

【0455】ステップ S515 において、チケットタイプがファイル登録チケット (FRT : File Registration Ticket) でないと判定された場合は、ステップ S519 においてチケットタイプを検証しチケットがサービス許可チケット (SPT : Service Permission Ticket) であるか否かを判定する。

【0456】チケットタイプがサービス許可チケット (SPT : Service Permission Ticket) である場合は、ステップ S520 ~ S522 を実行し、サービス許可チケット (SPT : Service Permission Ticket) でない場合は、ステップ S523 に進む。

【0457】チケットタイプがサービス許可チケット (SPT: Service Permission Ticket) である場合は、ステップS520において、チケットに記述された Integrity Check Type (チケット (Ticket) の正当性検証値の種別 (公開鍵方式(Public)/共通鍵方式(Common))) の設定が公開鍵方式(Public)であるか否かを判定する。

【0458】正当性検証値の種別 (Integrity Check Type) が公開鍵方式(Public)である場合、ステップS521に進み、各種処理を実行する。ステップS521で実行する処理は、まず、パーティションマネージャ対応認証局 (CA (PAR)) の公開鍵 PUB CA (PAR) を用いたチケット発行者 (Ticket Issuer) の公開鍵証明書 (CERT) の検証処理である。

【0459】サービス許可チケット (SPT) 発行手段 (SPT Issuer) の発行したチケット (Ticket) を、チケットユーザに対して送信する際には、公開鍵方式の場合、サービス許可チケット (SRT) 発行手段 (SPT Issuer) の公開鍵証明書 (CERT\_SPTI) も一緒にデバイスに送信される。なお、SPT発行手段の公開鍵証明書 (CERT\_SPTI) の属性 (Attribute) は、サービス許可チケット (SPT) 発行手段 (SPT Issuer) の識別子 (SPT IC) と一致する。

【0460】公開鍵証明書 (図11参照) にはパーティションマネージャ対応認証局 (CA (PAR)) の秘密鍵で実行された署名が付加されており、この署名をパーティションマネージャ対応認証局 (CA (PAR)) の公開鍵 PUB CA (PAR) を用いて検証する。署名生成、検証は、例えば先に説明した図12、図13のフローに従った処理として実行される。この署名検証により、チケット発行者の公開鍵証明書 (CERT) が改竄されたものでない正当な公開鍵証明書 (CERT) であるか否かが判定される。

【0461】さらに、ステップS521では、署名検証により正当性が確認されたチケット発行手段の公開鍵証明書 (CERT) のオプション領域に記録されたユーザの所属コードと、デバイス内のファイル定義ブロック (FDB: File Definition Block) に記録されたチケット発行手段コード (SPT IC: SPT Issuer Code) と一致するか否かを判定する。

【0462】公開鍵証明書には、図11の公開鍵証明書の説明の欄で記述したように、各チケット (PRT, FRT, SPTなど) の発行手段であるチケット発行手段 (Ticket Issuer) の所属コード、この場合、SPT IC (SPT Issuer Code) が記録されている。このオプション領域のコードとデバイス内のFDB (File Definition Block) に記録されたチケット発行手段コード (SPT IC: SPT Issuer Code) の一致を確認することで、受信チケット (SPT) が正当なチケット発行手段によって発行されたチケットであることを確認する。

【0463】さらに、デバイスは、デバイスのメモリ部内のパーティション鍵領域 (図23参照) に格納された CRL\_PAR (排除デバイス (Device)、排除機器 (デバイスアクセス機器としてのリーダライタ、PC等のチケットユーザ、チケット発行手段) の公開鍵証明書識別子 (ex. シリアルナンバ: SN) を登録したリボケーションリスト (Revocation List (Certificate))) を参照して、チケット発行手段 (Ticket Issuer) がリボークされていないかを判定する。

【0464】さらに、受信チケットであるサービス許可チケット (SPT) (図28、図31参照) に記録された署名、すなわち Integrity Check Value (チケット (Ticket) の正当性検証値 (公開鍵方式: 署名 (Signature)) の検証を実行し、チケットが改竄されていないかを確認する。署名検証は、先の公開鍵証明書の署名検証と同様、例えば図13のフローと同様のシーケンスに従って実行される。

【0465】以上、(1) チケット発行者 (Ticket Issuer) の公開鍵証明書 (CERT) が改竄されたものでない正当な公開鍵証明書 (CERT) であること、

(2) チケット発行者 (Ticket Issuer) の公開鍵証明書 (CERT) のオプション領域に記録されたコードと、デバイス内のFDB (File Definition Block) に記録されたチケット発行手段コード (SPT IC: SPT Issuer Code) の一致、(3) チケット発行手段 (Ticket Issuer) がリボークされていないこと、(4) 受信チケット (SPT) の署名 (Signature) の検証によりチケットが改竄のないことの確認。以上のすべての確認がなされたことを条件としてサービス許可チケット (SPT) の正当性検証成功とする。上記(1)～(4)のいずれかが確認されない場合は、サービス許可チケット (SPT) の正当性の確認が得られないと判定され、サービス許可チケット (SPT) を利用した処理は中止される。

【0466】また、ステップS520において、チケットに記述された Integrity Check Type (チケット (Ticket) の正当性検証値の種別 (公開鍵方式(Public)/共通鍵方式(Common))) の設定が共通鍵方式(Common)であると判定された場合は、ステップS522に進みMAC (Message Authentication Code) 検証を実行する。デバイスは、デバイスのファイル定義ブロック (図24参照) に格納されたサービス許可チケット (SPT) のMAC検証用鍵: K<sub>spt</sub>を使用してチケットのMAC検証処理を実行する。MAC検証処理は、先に説明した図59のDES暗号処理構成を用いたMAC値生成処理に従って実行される。

【0467】改竄のないことが保証された例えばデータ送信側がデータ生成時に生成したICV (Integrity Check Value) と、データ受信側が受信データに基づいて生成したICVとを比較して同一のICVが得られれば

データに改竄のないことが保証され、ICVが異なれば、改竄があったと判定される。改竄のないことが保証された例えばデータ送信側がデータ生成時に生成したICVは、図28、図31のサービス許可チケット(SPT)のフォーマットに関する記述において説明したように、SPTのICV(Integrity Check Value)フィールドに格納されている。デバイスが生成したICVと受信チケット(SPT)に格納されたICVとを比較して一致していればチケットの正当性ありと判定し、不一致の場合はチケット改竄ありと判定し、サービス許可チケット(SPT)を利用した処理を中止する。

【0468】上述の処理によってサービス許可チケット(SPT)に記述されたIntegrity Check Typeが共通鍵方式である場合のサービス許可チケット(SPT)検証処理が完了する。

【0469】ステップS519において、チケットタイプがサービス許可チケット(SPT: Service Permission Ticket)でないと判定された場合は、ステップS523においてチケットタイプを検証しチケットがデータアップデートチケット-DEV(DUT: Data Update Ticket(DEV)) (図32参照)であるか否かを判定する。データアップデートチケット(DUT)は前述したようにデバイスのメモリ部に格納された各種データの更新処理を実行する際のアクセス許可チケットであり、デバイスマネージャの管理データを更新する処理に適用するデータアップデートチケット-DEV(DUT(DEV))とパーティションマネージャの管理データを更新する処理に適用するデータアップデートチケット-PAR(DUT(PAR))とがある。

【0470】チケットタイプがデータアップデートチケット-DEV(DUT(DEV))である場合は、ステップS524~S528を実行し、データアップデートチケット(DEV)(DUT: Data Update Ticket(DEV))でない場合は、ステップS529に進む。

【0471】チケットタイプがデータアップデートチケット-DEV(DUT(DEV))である場合は、ステップS524において、チケットに記述されたIntegrity Check Type(チケット(Ticket)の正当性検証値の種類(公開鍵方式(Public)/共通鍵方式(Common)))の設定が公開鍵方式(Public)であるか否かを判定する。

【0472】正当性検証値の種類(Integrity Check Type)が公開鍵方式(Public)である場合、ステップS525に進み、各種処理を実行する。ステップS525で実行する処理は、まず、デバイスマネージャ対応認証局(CA(DEV))の公開鍵PUB CA(DEV)を用いたチケット発行者(Ticket Issuer)の公開鍵証明書(CERT)の検証処理である。

【0473】データアップデートチケット-DEV(DUT(DEV))発行手段(DUT Issuer)の発行したチケット(Ticket)を、チケットユーザに対して送信する

際には、公開鍵方式の場合、データアップデートチケット(DUT)発行手段(DUT Issuer)の公開鍵証明書(CERT\_DUTI)も一緒にデバイスに送信される。なお、DUT発行手段の公開鍵証明書(CERT\_DUTI)の属性(Attribute)は、デバイス内のDKDB(PUB)(Device Key Definition Block)(PUB)に記録されたチケット発行手段コード(DUTIC\_DEV)の識別子(DUTIC)と一致する。

【0474】公開鍵証明書(図11参照)にはデバイスマネージャ対応認証局(CA(DEV))の秘密鍵で実行された署名が付加されており、この署名をデバイスマネージャ対応認証局(CA(DEV))の公開鍵PUB CA(DEV)を用いて検証する。署名生成、検証は、例えば先に説明した図12、図13のフローに従った処理として実行される。この署名検証により、チケット発行者の公開鍵証明書(CERT)が改竄されたものではない正当な公開鍵証明書(CERT)であるか否かが判定される。

【0475】さらに、ステップS525では、署名検証により正当性が確認されたチケット発行手段の公開鍵証明書(CERT)のオプション領域に記録されたユーザの所属コードと、デバイス内のDKDB(PUB)(Device Key Definition Block)(PUB)に記録されたチケット発行手段コード(DUTIC\_DEV: DUT Issuer Category for Device)と一致するか否かを判定する。

【0476】公開鍵証明書には、図11の公開鍵証明書の説明の欄で記述したように、各チケット(PRT, FRT, SPT, DUT)の発行手段であるチケット発行手段(Ticket Issuer)の所属コード、この場合、DUTIC(DUT Issuer Code)が記録されている。このオプション領域のコードとデバイス内のDKDB(PUB)(Device Key Definition Block)(PUB)に記録されたチケット発行手段コード(DUTIC\_DEV: DUT Issuer Category for Device)(図16参照)の一致を確認することで、受信チケット(DUT)が正当なチケット発行手段によって発行されたチケットであることを確認する。

【0477】さらに、デバイスは、デバイスのメモリ部内のデバイス鍵領域(図18参照)に格納されたCRL\_DEV(排除デバイス(Device)、排除機器(デバイスアクセス機器としてのリーダライタ、PC等のチケットユーザ、チケット発行手段)の公開鍵証明書識別子(ex. シリアルナンバ: SN)を登録したリボケーションリスト(Revocation List(Certificate)))を参照して、チケット発行手段(Ticket Issuer)がリボークされていないかを判定する。

【0478】さらに、受信チケットであるデータアップデートチケット-DEV(DUT(DEV)) (図32参照)に記録された署名、すなわちIntegrity Check Value(チケット(Ticket)の正当性検証値(公開鍵方式: 署名(Signature))の検証を実行し、チケットが



改竄されていないかを確認する。署名検証は、先の公開鍵証明書の署名検証と同様、例えば図13のフローと同様のシーケンスに従って実行される。

【0479】以上、(1) チケット発行者 (Ticket Issuer) の公開鍵証明書 (CERT) が改竄されたものでない正当な公開鍵証明書 (CERT) であること、

(2) チケット発行者 (Ticket Issuer) の公開鍵証明書 (CERT) のオプション領域に記録されたコードと、デバイス内のDKDB (PUB) (Device Key Definition Block) (PUB)に記録されたチケット発行手段コード (DUTIC\_DEV: DUT Issuer Category for Device) の一致、(3) チケット発行手段 (Ticket Issuer) がリポーカされていないこと、(4) 受信チケット (DUT) の署名 (Signature) の検証によりチケットが改竄のないことの確認。以上のすべての確認がなされたことを条件としてデータアップデートチケット-DEV (DUT (DEV)) の正当性検証成功とする。上記(1)～(4)のいずれかが確認されない場合は、データアップデートチケット-DEV (DUT (DEV)) の正当性の確認が得られないと判定され、データアップデートチケット-DEV (DUT (DEV)) を利用した処理は中止される。

【0480】また、ステップS524において、チケットに記述されたIntegrity Check Type (チケット (Ticket) の正当性検証値の種別 (公開鍵方式 (Public) / 共通鍵方式 (Common)) ) の設定が共通鍵方式 (Common) であると判定された場合は、ステップS526において、データアップデートチケット-DEV (DUT (DEV)) に記述されたOld Data Codeの示すデータがデバイス鍵領域 (図18参照) に格納されたKdut\_DEV1 (データアップデートチケット (DUT) のMAC検証用鍵) または、Kdut\_DEV2 (データ更新用暗号鍵) であるか否かを判定する。

【0481】データアップデートチケット-DEV (DUT (DEV)) に記述されたOld Data Code (更新される古いデータのコード) の示すデータがデバイス鍵領域 (図18参照) に格納されたKdut\_DEV1 (データアップデートチケット (DUT) のMAC検証用鍵) または、Kdut\_DEV2 (データ更新用暗号鍵) である場合は、ステップS528において、デバイス鍵領域 (図18参照) に格納されたKdut\_DEV3 (データアップデートチケット (DUT) のMAC検証用鍵) を用いてMAC検証処理を実行し、データアップデートチケット-DEV (DUT (DEV)) に記述されたOld Data Code (更新される古いデータのコード) の示すデータがデバイス鍵領域 (図18参照) に格納されたKdut\_DEV1 (データアップデートチケット (DUT) のMAC検証用鍵) または、Kdut\_DEV2 (データ更新用暗号鍵) でない場合は、ステップS527において、デバイス鍵領域 (図18参照) に格納されたKdut\_DEV1 (データアップデート

チケット (DUT) のMAC検証用鍵) を用いてMAC検証処理を実行する。

【0482】上述のようにMAC検証鍵の使い分けを実行するのは、更新対象となっているデータが、Kdut\_DEV1 (データアップデートチケット (DUT) のMAC検証用鍵) または、Kdut\_DEV2 (データ更新用暗号鍵) である場合は、これらの鍵データが何らかの理由、例えば鍵情報の漏洩等により、使用を停止される予定の情報であるため、これらの更新対象データを用いたMAC検証を避けるためである。MAC検証処理は、先に説明した図59のDES暗号処理構成を用いたMAC値生成処理に従って実行される。

【0483】なお、デバイスは、デバイスのデバイス鍵領域 (図18参照) に新規にKdut\_DEV1 (データアップデートチケット (DUT) のMAC検証用鍵) を格納する場合、以前に格納済みのKdut\_DEV3 (データアップデートチケット (DUT) のMAC検証用鍵) とのスワップ、すなわち入れ替え処理を行なう。さらに、新規にKdut\_DEV2 (データ更新用暗号鍵) を格納する場合も、以前に格納済みのKdut\_DEV4 (データ更新用暗号鍵) とのスワップ、すなわち入れ替え処理を行なう。

【0484】この、Kdut\_DEV1と、Kdut\_DEV3とのスワップ、および、Kdut\_DEV2と、Kdut\_DEV4とのスワップ処理によって、常にKdut\_DEV3 (データアップデートチケット (DUT) のMAC検証用鍵)、Kdut\_DEV4 (データ更新用暗号鍵) のペアがKdut\_DEV1 (データアップデートチケット (DUT) のMAC検証用鍵)、Kdut\_DEV2 (データ更新用暗号鍵) のペアよりも新しいバージョンのものに維持される。つまり、Kdut\_DEV1と、Kdut\_DEV2の鍵は常に使用される鍵で、Kdut\_DEV3と、Kdut\_DEV4は、非常時にKdut\_DEV1と、Kdut\_DEV2を更新するとともに、現在使用されているKdut\_DEV1と、Kdut\_DEV2の鍵に置き換えられるバックアップ用の鍵としての役割がある。なお、これらの処理については、後段のデータアップデートチケット (DUT) を用いたデータ更新処理の説明において、さらに説明する。

【0485】改竄のないことが保証された例えばデータ送信側がデータ生成時に生成したICV (Integrity Check Value) と、データ受信側が受信データに基づいて生成したICVとを比較して同一のICVが得られればデータに改竄のないことが保証され、ICVが異なれば、改竄があったと判定される。改竄のないことが保証された例えばデータ送信側がデータ生成時に生成したICVは、図32のデータアップデートチケット (DUT) のフォーマットに関する記述において説明したように、データアップデートチケット (DUT) のICV (Integrity Check Value) フィールドに格納されている。

【0486】デバイスが生成したICVと受信チケットであるデータアップデートチケット-DEV (DUT

(DEV))に格納されたICVとを比較して一致していればチケットの正当性ありと判定し、不一致の場合はチケット改竄ありと判定し、データアップデートチケット-DEV(DUT(DEV))を利用した処理を中止する。

【0487】上述の処理によってデータアップデートチケット-DEV(DUT(DEV))に記述されたIntegrity Check Typeが共通鍵方式である場合のデータアップデートチケット-DEV(DUT(DEV))検証処理が完了する。

【0488】ステップS523において、チケットタイプがデータアップデートチケット-DEV(DUT(DEV))でないとは判定された場合は、チケットは、データアップデートチケット-PAR(DUT(PAR)) (図32参照)であると判定される。データアップデートチケット-PAR(DUT(PAR))は、パーティションマネージャの管理データを更新する処理に適用するチケットである。

【0489】この場合、ステップS529において、チケットに記述されたIntegrity Check Type(チケット(Ticket)の正当性検証値の種別(公開鍵方式(Public)/共通鍵方式(Common))の設定が公開鍵方式(Public)であるか否かを判定する。

【0490】正当性検証値の種別(Integrity Check Type)が公開鍵方式(Public)である場合、ステップS530に進み、各種処理を実行する。ステップS530で実行する処理は、まず、パーティションマネージャ対応認証局(CA(PAR))の公開鍵PUB CA(PAR)を用いたチケット発行者(Ticket Issuer)の公開鍵証明書(CERT)の検証処理である。

【0491】データアップデートチケット-PAR(DUT(PAR))発行手段(DUT Issuer)の発行したチケット(Ticket)を、チケットユーザに対して送信する際には、公開鍵方式の場合、データアップデートチケット(DUT)発行手段(DUT Issuer)の公開鍵証明書(CERT\_DUTI)も一緒にデバイスに送信される。なお、DUT発行手段の公開鍵証明書(CERT\_DUTI)の属性(Attribute)は、デバイス内のPKDB(PUB)(Partition Key Definition block)に記録されたチケット発行手段コード(DUTIC\_PAR)と一致する。

【0492】公開鍵証明書(図11参照)にはパーティションマネージャ対応認証局(CA(PAR))の秘密鍵で実行された署名が付加されており、この署名をパーティションマネージャ対応認証局(CA(PAR))の公開鍵PUB CA(PAR)を用いて検証する。署名生成、検証は、例えば先に説明した図12、図13のフローに従った処理として実行される。この署名検証により、チケット発行者の公開鍵証明書(CERT)が改竄されたものでない正当な公開鍵証明書(CERT)であるか否かが判定される。

【0493】さらに、ステップS530では、署名検証により正当性が確認されたチケット発行手段の公開鍵証明書(CERT)のオプション領域に記録されたユーザの所属コードと、デバイス内のPKDB(PUB)(Partition Key Definition block)に記録されたチケット発行手段コード(DUTIC\_PAR:DUT Issuer Category for Partition)と一致するか否かを判定する。

【0494】公開鍵証明書には、図11の公開鍵証明書の説明の欄で記述したように、各チケット(PRT, FRT, SPT, DUT)の発行手段であるチケット発行手段(Ticket Issuer)の所属コード、この場合、DUTIC(DUT Issuer Code)が記録されている。このオプション領域のコードとデバイス内のPKDB(PUB)(Partition Key Definition block)に記録されたチケット発行手段コード(DUTIC:DUT Issuer Category)(図21参照)の一致を確認することで、受信チケット(DUT)が正当なチケット発行手段によって発行されたチケットであることを確認する。

【0495】さらに、デバイスは、デバイスのメモリ部内のデバイス鍵領域(図18参照)に格納されたCRL\_DEV(排除デバイス(Device)、排除機器(デバイスアクセス機器としてのリーダライタ、PC等のチケットユーザ、チケット発行手段)の公開鍵証明書識別子(ex. シリアルナンバ:SN)を登録したリボケーションリスト(Revocation List(Certificate))を参照して、チケット発行手段(Ticket Issuer)がリボークされていないかを判定する。

【0496】さらに、受信チケットであるデータアップデートチケット-PAR(DUT(PAR)) (図32参照)に記録された署名、すなわちIntegrity Check Value(チケット(Ticket)の正当性検証値(公開鍵方式:署名(Signature))の検証を実行し、チケットが改竄されていないかを確認する。署名検証は、先の公開鍵証明書の署名検証と同様、例えば図13のフローと同様のシーケンスに従って実行される。

【0497】以上、(1)チケット発行者(Ticket Issuer)の公開鍵証明書(CERT)が改竄されたものでない正当な公開鍵証明書(CERT)であること、

(2)チケット発行者(Ticket Issuer)の公開鍵証明書(CERT)のオプション領域に記録されたコードと、デバイス内のPKDB(PUB)(Partition Key Definition block)に記録されたチケット発行手段コード(DUTIC\_PAR:DUT Issuer Category for Partition)の一致、(3)チケット発行手段(Ticket Issuer)がリボークされていないこと、(4)受信チケット(DUT)の署名(Signature)の検証によりチケットが改竄のないことの確認。以上のすべての確認がなされたことを条件としてデータアップデートチケット-PAR(DUT)の正当性検証成功とする。上記(1)~(4)のいずれかが確認されない場合は、データアップデートチ

ケット-PAR (DUT (PAR)) の正当性の確認が得られないと判定され、データアップデートチケット-PAR (DUT (PAR)) を利用した処理は中止される。

【0498】また、ステップS529において、チケットに記述されたIntegrity Check Type (チケット (Ticket) の正当性検証値の種別 (公開鍵方式(Public) /共通鍵方式(Common)) ) の設定が共通鍵方式(Common)であると判定された場合は、ステップS531において、データアップデートチケット-PAR (DUT (PAR)) に記述されたOld Data Code (更新される古いデータのコード) の示すデータがパーティション鍵領域 (図23参照) に格納されたKdut\_PAR1 (データアップデートチケット (DUT) のMAC検証用鍵) または、Kdut\_PAR2 (データ更新用暗号鍵) であるか否かを判定する。

【0499】データアップデートチケット-PAR (DUT (PAR)) に記述されたOld Data Code (更新される古いデータのコード) の示すデータがパーティション鍵領域 (図23参照) に格納されたKdut\_PAR1 (データアップデートチケット (DUT) のMAC検証用鍵) または、Kdut\_PAR2 (データ更新用暗号鍵) である場合は、ステップS533において、パーティション鍵領域 (図23参照) に格納されたKdut\_PAR3 (データアップデートチケット (DUT) のMAC検証用鍵) を用いてMAC検証処理を実行し、データアップデートチケット-PAR (DUT (PAR)) に記述されたOld Data Code (更新される古いデータのコード) の示すデータがパーティション鍵領域 (図23参照) に格納されたKdut\_PAR1 (データアップデートチケット (DUT) のMAC検証用鍵) または、Kdut\_PAR2 (データ更新用暗号鍵) でない場合は、ステップS532において、パーティション鍵領域 (図23参照) に格納されたKdut\_PAR1 (データアップデートチケット (DUT) のMAC検証用鍵) を用いてMAC検証処理を実行する。

【0500】上述のようにMAC検証鍵の使い分けを実行するのは、更新対象となっているデータが、Kdut\_PAR1 (データアップデートチケット (DUT) のMAC検証用鍵) または、Kdut\_PAR2 (データ更新用暗号鍵) である場合は、これらの鍵データが何らかの理由、例えば鍵情報の漏洩等により、使用を停止される予定の情報であるため、これらの更新対象データを用いたMAC検証を避けるためである。MAC検証処理は、先に説明した図59のDES暗号処理構成を用いたMAC値生成処理に従って実行される。

【0501】改竄のないことが保証された例えばデータ送信側がデータ生成時に生成したICV (Integrity Check Value) と、データ受信側が受信データに基づいて生成したICVとを比較して同一のICVが得られればデータに改竄のないことが保証され、ICVが異なれば、改竄があったと判定される。改竄のないことが保証

された例えばデータ送信側がデータ生成時に生成したICVは、図32のデータアップデートチケット (DUT) のフォーマットに関する記述において説明したように、データアップデートチケット (DUT) のICV (Integrity Check Value) フィールドに格納されている。

【0502】デバイスが生成したICVと受信チケットであるデータアップデートチケット-PAR (DUT (PAR)) に格納されたICVとを比較して一致していればチケットの正当性ありと判定し、不一致の場合はチケット改竄ありと判定し、データアップデートチケット-PAR (DUT (PAR)) を利用した処理を中止する。

【0503】上述の処理によってデータアップデートチケット-PAR (DUT (PAR)) に記述されたIntegrity Check Typeが共通鍵方式である場合のデータアップデートチケット-PAR (DUT (PAR)) 検証処理が完了する。

【0504】以上の処理においてチケットの正当性が確認された後、図58のステップS541に進み、以下、利用者チェック、すなわちチケットユーザとしてデバイスとの通信を実行中のデバイスアクセス機器としてのリーダライタ (またはPC等) のチェックを実行する。

【0505】ステップS541において、デバイスは、受信チケット (PRT, FRT, SPT, またはDUT) のAuthentication Flag (チケット (Ticket) の利用処理においてデバイス (Device) との相互認証が必要か否かを示すフラグ) をチェックする。フラグが認証不要を示している場合は、処理を実行することなく終了する。

【0506】ステップS541におけるフラグチェック処理において、フラグが認証必要を示している場合は、ステップS542に進み、チケットユーザ (デバイスに対するチケットを適用した処理を実行しようとするデバイスアクセス機器としてのリーダライタ、PC等) の所属 (グループ) をキーとして認証テーブル (図51参照) を参照する。

【0507】次に、ステップS543において、受信チケットのAuthentication Type (デバイス (Device) の相互認証のタイプ (公開鍵認証、または、共通鍵認証、または、いずれでも可 (Any)) を記録したデータ) をチェックし、いずれでも可 (Any) である場合、ステップS544に進み、ステップS542でチェックしたグループの相互認証データが認証テーブル (図51参照) に格納されているか否かを判定する。テーブルに対応グループの相互認証情報が格納され、チケットユーザ (デバイスに対するチケットを適用した処理を実行しようとするデバイスアクセス機器としてのリーダライタ、PC等) とデバイス間の相互認証済みであることが判定されれば、チケット利用者 (ex. デバイスアクセス機器と

してのリーダライタ)の正当性が確認されたものとして処理を利用者チェック成功と判定して終了する。認証テーブル(図51参照)に対応グループの相互認証情報が格納されていない場合は、利用者チェックが未了であると判定され、エラー終了とする。

【0508】ステップS543において、受信チケットのAuthentication Type(デバイス(Device)の相互認証のタイプ(公開鍵認証、または、共通鍵認証、または、いずれでも可(Any))を記録したデータ)がいずれでも可(Any)でない場合、ステップS545において、Authentication Typeが公開鍵認証であるか否かを判定する。

【0509】Authentication Typeが公開鍵認証である場合、ステップS546に進み、ステップS542でチェックしたグループの公開鍵相互認証データが認証テーブル(図51参照)に格納されているか否かを判定する。テーブルに対応グループの公開鍵相互認証情報が格納され、チケットユーザ(デバイスに対するチケットを適用した処理を実行しようとするデバイスアクセス機器としてのリーダライタ、PC等)とデバイス間の相互認証が公開鍵認証処理として成立済みであることが判定された場合は、ステップS547に進み、処理対象チケット(PRT, FRT, SPTまたはDUT)にチケットユーザの識別子が存在するか否かを判定して存在する場合は、ステップS548において認証相手(チケットユーザであるデバイスアクセス機器としてのリーダライタなど)の公開鍵証明書中の識別データ(DN)として記録された識別子またはカテゴリまたはシリアル(SN)とチケットに格納されたチケットユーザの識別データとして記録された識別子またはカテゴリまたはシリアル(SN)が一致するか否かを判定する。一致する場合は、利用者確認成功として処理を終了する。

【0510】ステップS546において、ステップS542でチェックしたグループの公開鍵相互認証データが認証テーブル(図51参照)に格納されておらず、チケットユーザ(デバイスに対するチケットを適用した処理を実行しようとするデバイスアクセス機器としてのリーダライタ、PC等)とデバイス間の相互認証が公開鍵認証処理として成立済みでないとは判定された場合は、利用者チェック未了と判定されエラー終了する。

【0511】また、ステップS548において認証相手(チケットユーザであるデバイスアクセス機器としてのリーダライタなど)の公開鍵証明書中の識別データ(DN)として記録された識別子またはカテゴリまたはシリアル(SN)とチケットに格納されたチケットユーザの識別子が一致しないと判定された場合も利用者チェック未了と判定されエラー終了する。

【0512】なお、チケットにチケットユーザの識別子が存在しない場合は、ステップS548の処理は実行せず、利用者確認成功として処理を終了する。

【0513】ステップS545において、受信チケットのAuthentication Type(デバイス(Device)の相互認証のタイプ(公開鍵認証、または、共通鍵認証、または、いずれでも可(Any))を記録したデータ)が公開鍵認証でないとは判定された場合、ステップS549に進み、ステップS542でチェックしたグループの共通鍵相互認証データが認証テーブル(図51参照)に格納されているか否かを判定する。テーブルに対応グループの共通鍵相互認証情報が格納され、チケットユーザ(デバイスに対するチケットを適用した処理を実行しようとするデバイスアクセス機器としてのリーダライタ、PC等)とデバイス間の相互認証が共通鍵認証処理として成立済みであることが判定された場合は、ステップS550に進み、処理対象チケット(PRT, FRT, SPTまたはDUT)にチケットユーザの識別子が存在するか否かを判定して存在する場合は、ステップS551において認証相手(チケットユーザであるデバイスアクセス機器としてのリーダライタなど)の識別データ(IDr w)とチケットに格納されたチケットユーザの識別子が一致するか否かを判定する。一致する場合は、利用者確認成功として処理を終了する。

【0514】ステップS549において、ステップS542でチェックしたグループの共通鍵相互認証データが認証テーブル(図51参照)に格納されておらず、チケットユーザ(デバイスに対するチケットを適用した処理を実行しようとするデバイスアクセス機器としてのリーダライタ、PC等)とデバイス間の相互認証が共通鍵認証処理として成立済みでないとは判定された場合は、利用者チェック未了と判定されエラー終了する。

【0515】また、ステップS551において認証相手(チケットユーザであるデバイスアクセス機器としてのリーダライタなど)の識別データ(IDr w)とチケットに格納されたチケットユーザの識別子が一致しないと判定された場合も利用者チェック未了と判定されエラー終了する。

【0516】なお、チケットにチケットユーザの識別子が存在しないか、すべてのチケットユーザが利用可能の場合は、ステップS550の処理は実行せず、利用者確認成功として処理を終了する。

【0517】以上が、図47のフロー中のステップS413においてデバイスが実行するチケットの正当性および利用者チェック処理である。

【0518】(パーティション作成削除処理)次に、図47のフローに示すステップS415において実行されるパーティション登録チケット(PRT)に基づくパーティションの生成、削除処理の詳細について、図60、図61の処理フローを用いて説明する。パーティションの作成、削除処理は、チケットユーザ(ex. デバイスアクセス機器としてのリーダライタ、PC等)からパーティション登録チケット(PRT)を受信したデバイス

が、パーティション登録チケット (PRT) に基づいて実行する処理である。

【0519】図60のステップS601において、デバイスは、受信したパーティション登録チケット (PRT: Partition Registration ticket) に記録された処理タイプ、すなわち Operation Type (パーティション (Partition) 作成か削除かの指定 (作成 (Generate) / 削除 (Delete)) ) を検証する。処理タイプ (Operation Type) が、パーティション (Partition) 作成である場合、ステップS602以下を実行し、パーティション (Partition) 削除である場合、ステップS621以下を実行する。

【0520】まず、パーティション作成処理について説明する。デバイスはステップS602において、パーティション登録チケット (PRT) に記述されたパーティションマネージャコード (PMC) と同一コードのパーティションがデバイスのメモリ部に存在するか否かを検証する。この判定は、デバイスのメモリ部のパーティション定義ブロック (図19参照) に受信チケット (PRT) の記述コードと同一のコードが記述されているか否かを検証することによって判定可能である。

【0521】すでにデバイスに同一コード (PMC) のパーティションが存在する場合は、同一コードを持つ重複パーティションの存在は許されず、パーティションの生成は実行せず、エラー終了とする。同一コードのパーティションがデバイスに存在しない場合は、ステップS603において、デバイス管理情報ブロック (図15参照) のデバイス (Device) 内の空きブロック数 (Free Block Number in Device) と、パーティション登録チケット (PRT) に記述されたパーティションサイズ (Partition Size) とを比較し、チケット (PRT) に記述されたパーティションサイズ (Partition Size) 以上の空きブロック領域がデバイスのメモリ部に存在するか否かを判定する。存在しない場合は、PRTに記述されたサイズのパーティションの生成はできないので、エラー終了とする。

【0522】チケット (PRT) に記述されたパーティションサイズ (Partition Size) 以上の空きブロック領域がデバイスのメモリ部に存在すると判定された場合は、ステップS604に進み、デバイス管理情報ブロックの空き領域ポインタ (Pointer of Free Area) を参照してデバイスの空き領域 (Free Area in Device) の最上位ブロックにパーティション定義ブロック (PDB) エリア (図19参照) を確保する。

【0523】次に、デバイスは、確保したパーティション定義ブロック (PDB) エリアに、パーティション登録チケット (PRT) に記述されたパーティションマネージャコード (PMC) のコピー (S605)、PRTに記述されたPMCバージョンのコピー、(S606) を実行する。

【0524】さらに、パーティション定義ブロック (PDB) エリアのパーティションスタート位置 (Partition Start Position) に、デバイス管理情報ブロック (図15参照) の空き領域ポインタ (Pointer of Free Area) のコピー処理を実行 (S607) し、さらに、パーティション定義ブロック (PDB) エリアのパーティションサイズ (Partition Size) にパーティション登録チケット (PRT) に記述されたパーティションサイズ (Partition Size) のコピー処理を実行 (S608) する。

【0525】次に、デバイス管理情報ブロック (図15参照) の空き領域ポインタ (Pointer of Free Area) にパーティション定義ブロック (PDB) エリアのパーティションサイズ (Partition Size) にコピーした値を加算 (S609) し、デバイス管理情報ブロック (図15参照) のデバイス (Device) 内の空きブロック数 (Free Block Number in Device) からパーティションサイズ (Partition Size) + 1 を減算する (S610)。なお、+1 は、パーティション定義ブロック (PDB) 用のブロックを意味する。

【0526】次にデバイス管理情報ブロック (図15参照) のパーティション数 (Partition Number) に1を加算、すなわち生成したパーティション数 (1) を加算する (S611)。

【0527】次に、図61のステップS631において、生成したパーティション領域の最上位ブロックをパーティション管理情報ブロック (PMIB: partition Management Information Block) (図20参照) として設定し、設定したパーティション管理情報ブロック (PMIB) のパーティションマネージャコード (PMC) フィールドにパーティション登録チケット (PRT) のPMCのコピー処理を実行 (S632) し、パーティション管理情報ブロック (PMIB) のPMCバージョンフィールドにパーティション登録チケット (PRT) のPMCバージョンのコピー処理を実行 (S633) し、パーティション管理情報ブロック (PMIB) のパーティション総ブロック数 (Total Block number in Partition) フィールドにパーティション登録チケット (PRT) のパーティションサイズ (Partition Size) のコピー処理を実行 (S634) する。

【0528】さらに、パーティション管理情報ブロック (PMIB) のパーティション空きブロック数 (Free Block number in Partition) フィールドにパーティション登録チケット (PRT) のパーティションサイズ (Partition Size) - 3 を記録 (S635) する。なお、-3 の意味は、既に使用が予定されているパーティション管理情報ブロック (PMIB)、共通鍵系パーティション鍵定義ブロック (PKDB (common))、公開鍵系パーティション鍵定義ブロック (PKDB (PUB)) の3ブロックを差し引くことを意味している。

【0529】さらに、パーティション管理情報ブロック

(PMIB)のファイル数(File Number)に0を記入(S636)する。この時点ではパーティション内にはファイルは設定されていない。ファイル設定はファイル登録チケット(FRT)を使用して設定可能である。このファイル登録チケット(FRT)を使用したファイル登録処理については後述する。

【0530】さらに、パーティション管理情報ブロック(PMIB)の空き領域ポインタ(Pointer of Free Area)にパーティション定義ブロック(PDB)のスタートポジション(Start Position)をコピーしてパーティションの設定登録を終了する。

【0531】次に図60のステップS621～S628のパーティション削除処理について説明する。ステップS621では、パーティション登録チケット(PRT)に記述されたパーティションマネージャコード(PMC)と同一コードのパーティションがデバイスのメモリ部に存在するか否かを検証する。この判定は、デバイスのメモリ部のパーティション定義ブロック(図19参照)に受信チケット(PRT)の記述コードと同一のコードが記述されているか否かを検証することによって判定可能である。

【0532】デバイスに同一コード(PMC)のパーティションが存在しない場合は、パーティションの削除は不可能であるので、エラー終了とする。同一コードのパーティションがデバイスに存在する場合は、ステップS622において、削除対象のパーティションより後に生成されたパーティションがデバイスに存在するか否かを判定する。存在しない場合は、削除対象のパーティションが最新のパーティションであり、ステップS629において削除対象のパーティションのパーティション定義ブロック(PDB)(図19参照)を削除する。

【0533】ステップS622において、削除対象のパーティションより後に生成されたパーティションがデバイスに存在すると判定された場合は、後に生成されたパーティション(後パーティション)のデータを削除対象のパーティションのサイズ(PS)分、下位にずらす処理を実行(S623)し、さらに、後パーティションのパーティション定義ブロック(PDB)を1ブロック上位にずらす処理を実行(S624)する。また、後パーティションのパーティション定義ブロック(PDB)に記録されたパーティション開始位置(Partition Start Portion)から削除パーティションのサイズ(PS)を減算する処理を実行する(S625)。

【0534】ステップS625またはS629の処理の後、ステップS626において、デバイス管理情報ブロック(DMIB)(図15参照)のデバイス(Device)内の空きブロック数(Free Block Number in Device)に削除パーティションのサイズ(PS)+1を加算する。+1は、削除パーティションのパーティション定義ブロック(PDB)用のブロックを意味する。

【0535】次にステップS627において、デバイス管理情報ブロック(図15参照)の空き領域ポインタ(Pointer of Free Area)の値から削除パーティションのサイズ(PS)を減算する。さらに、ステップS628において、デバイス管理情報ブロック(図15参照)のパーティション数(Partition Number)から1を減算、すなわち削除したパーティション数(1)を減算してパーティション登録チケット(PRT)に基づくパーティション削除処理が終了する。

【0536】以上が、図47の処理フローにおけるステップS415のパーティション登録チケット(PRT)に基づくパーティション生成、削除処理である。

【0537】(パーティション初期登録)次に、図47の処理フローにおけるステップS406、S419のパーティション初期データ書き込み処理、すなわちパーティション登録チケット(PRT)に基づくパーティション初期登録処理の詳細について図62以下のフローを用いて説明する。

【0538】図62、図63、図64に示す処理フローにおいて、左側がパーティションマネージャの管轄する初期登録装置の処理、右側がデバイス(図5参照)の処理を示す。なお、パーティションマネージャの管轄する初期登録装置は、デバイスに対するデータ読み取り書き込み処理可能な装置(例えば、デバイスアクセス機器としてのリーダライタ、PC)であり、図10のデバイスアクセス機器としてのリーダライタに相当する構成を有する。図47の処理フローに示すように、図62の処理開始以前に、初期登録装置とデバイス間では、相互認証が成立し、チケットの正当性、利用者チェックにおいてチケットおよび利用者(チケットユーザであるデバイスアクセス機器としてのリーダライタなど)の正当性が確認され、さらにパーティション登録チケット(PRT)に基づくパーティション生成処理が終了しているものとする。また、図62、図63、図64の初期登録装置と、デバイス間のデータの送受信は、相互認証時に生成したセッション鍵Keysを用いて暗号化されたデータとして送受信される。

【0539】図62のステップS641において、初期登録装置は、パーティション認証に共通鍵を用いるか否かを判定する。この判定は、使用するパーティション登録チケット(PRT)(図26参照)のAuthentication Type(デバイス(Device)の相互認証のタイプ(公開鍵認証、または、共通鍵認証、または、いずれでも可(Any)))フィールドを参照して行われる。

【0540】図62に示すように、パーティション認証に共通鍵を用いる場合、ステップS642～S643、S651～S654を実行し、パーティション認証に共通鍵を用いない場合、これらのステップは省略される。

【0541】パーティション認証に共通鍵を用いる場合、ステップS642において初期登録装置は、共通鍵

認証データ書き込みコマンドとして、MKauth\_PAR\_A : 双方向個別鍵認証用マスター鍵、Kauth\_PAR\_B : 双方向個別鍵認証用共通鍵、IRL\_PAR: 排除デバイス (Device) のデバイス識別子 (ID) を登録したリボケーションリスト (Revocation List (Device ID))、およびこれらのバージョン情報をデバイスに送信する。

【0542】ステップS651でデバイスは、上述の書き込みコマンドを受信し、ステップS652において、受領データをパーティション鍵領域 (図23参照) に書き込む。次にデータ書き込みによって生じたポインタ、サイズ、デバイス内のフリーブロック数の調整を実行 (S653) し、書き込み終了通知を登録装置に送信 (S654) する。

【0543】書き込み終了通知を受信 (S643) した登録装置は、ステップS644においてパーティション認証に公開鍵を用いるか否かを判定する。図62に示すように、パーティション認証に公開鍵を用いる場合、ステップS645～649、S655～S662を実行し、パーティション認証に公開鍵を用いない場合、これらのステップは省略される。

【0544】パーティション認証に公開鍵を用いる場合、ステップS645において登録装置は、公開鍵認証データ書き込みコマンドとして、PUB\_CA (PAR) : パーティションマネージャ対応公開鍵証明書を発行する認証局 CA (PAR) の公開鍵、PARAM\_PAR : パーティション (Partition) の公開鍵パラメータ、CRL\_PAR : 排除デバイス (Device) の公開鍵証明書識別子 (ex. シリアルナンバ: SN) を登録したリボケーションリスト (Revocation List (Certificate))、およびこれらのバージョン情報をデバイスに送信する。

【0545】ステップS655でデバイスは、上述の書き込みコマンドを受信し、ステップS656において、受領データをパーティション鍵領域 (図23参照) に書き込む。次にデータ書き込みによって生じたポインタ、サイズ、デバイス内のフリーブロック数の調整を実行 (S657) し、書き込み終了通知を登録装置に送信 (S658) する。

【0546】書き込み終了通知を受信 (S646) した登録装置は、公開鍵と秘密鍵の鍵ペア生成コマンドをデバイスに送信 (S647) する。なお、この実施例では、鍵ペアの生成はデバイスが実行する構成としているが、例えば登録装置が実行してデバイスに提供する構成としてもよい。

【0547】鍵ペア生成コマンドを受信 (S659) したデバイスは、デバイス内の暗号処理部 (図5参照) において公開鍵 (PUB PAR) と秘密鍵 (PRI PAR) のペアを生成し、生成した鍵をパーティション鍵領域 (図23参照) に書き込む (S660)。次にデータ書き込みによって生じたポインタ、サイズ、デバイス内のフリーブロック数の調整を実行 (S661) し、生成

格納した公開鍵を登録装置に送信 (S662) する。

【0548】登録装置は、デバイスから公開鍵 (PUB PAR) を受信 (S648) し、先にデバイスから受信したデバイスの識別子 IDm とともに、パーティションマネージャ内のデータベース (DB (PAR)) (図9参照) に保存する。

【0549】次に、パーティションマネージャの登録装置は、ファイル登録チケット (FRT: File Registration Ticket) の検証処理に共通鍵を用いるか否かを判定 (S671) する。チケット検証には、前述したようにMAC値検証等による共通鍵方式と、秘密鍵による署名生成、公開鍵による署名検証を行なう公開鍵方式のいずれかを適用することが可能であり、パーティションマネージャは、デバイスの採用する検証処理方式を設定することができる。パーティションマネージャは、デバイスの採用するFRTチケット検証方式に応じて共通鍵、公開鍵のいずれか、あるいは両方式を実行可能なデータをデバイスに設定する。

【0550】パーティションマネージャは、ファイル登録チケット (FRT: File Registration Ticket) の検証処理に共通鍵認証を実行する設定とする場合は、共通鍵方式のFRT検証に必要な情報 (ex. FRT検証共通鍵) をデバイスにセットし、デバイスが共通鍵認証を実行しないデバイスであれば、これらの情報をデバイスに格納しないことになる。

【0551】図63に示すように、FRT検証に共通鍵方式を用いる場合、ステップS672～673、S681～S684を実行し、FRT検証に共通鍵を用いない場合、これらのステップは省略される。

【0552】FRT検証に共通鍵を用いる場合、ステップS672において登録装置は、FRT検証共通鍵書き込みコマンドとして、Kfrt : ファイル登録チケット (FRT) のMAC検証用鍵、およびバージョン情報をデバイスに送信する。

【0553】ステップS681でデバイスは、上述の書き込みコマンドを受信し、ステップS682において、受領データをパーティション鍵領域 (図23参照) に書き込む。次にデータ書き込みによって生じたポインタ、サイズ、デバイス内のフリーブロック数の調整を実行 (S683) し、書き込み終了通知を登録装置に送信 (S684) する。

【0554】書き込み終了通知を受信 (S673) した登録装置は、ステップS674においてFRT検証に公開鍵を用いるか否かを判定する。図63に示すように、FRT検証に公開鍵を用いる場合、ステップS675～676、S685～S690を実行し、FRT検証に公開鍵を用いない場合、これらのステップは省略される。

【0555】FRT検証に公開鍵を用いる場合、ステップS675において登録装置は、FRT検証データ書き込みコマンドとして、FRTIC (FRT Issuer Category) :

ファイル登録チケット (FRT) 発行者カテゴリ、PUB\_CA(PAR) :パーティションマネージャ対応公開鍵証明書  
を発行する認証局CA (PAR) の公開鍵、PARAM\_PAR  
:パーティション (Partition) の公開鍵パラメータ、  
CRL\_PAR :排除デバイス (Device) の公開鍵証明書識別  
子 (ex. シリアルナンバ : SN) を登録したリボケー  
ションリスト (Revocation List (Certificate))、お  
よびこれらのバージョン情報をデバイスに送信する。

【0556】ステップS685でデバイスは、上述の書  
き込みコマンドを受信し、ステップS686において、  
受領データ中のFRTIC (FRT Issuer Category) :ファイ  
ル登録チケット (FRT) 発行者カテゴリを公開鍵系パ  
ーティション鍵定義ブロック (PKDB : Partition Ke  
y Definition block (PUB) (図22参照)) に書き込  
みバージョン情報を同ブロックのバージョン領域に書き  
込む。

【0557】次にデバイスは、ステップS687におい  
て、PUB\_CA(PAR) :パーティションマネージャ対応公開  
鍵証明書を発行する認証局CA (PAR) の公開鍵デー  
タが書き込み済みか否かを判定し、書き込まれていない  
場合にステップS688において、PUB\_CA(PAR)、PARAM  
\_PAR、CRL\_PARをパーティション鍵領域 (図23参照)  
に書き込む。次にデータ書き込みによって生じたポイン  
タ、サイズ、デバイス内のフリーブロック数の調整を実  
行 (S689) し、書き込み終了通知を登録装置に送信  
(S690) する。

【0558】書き込み終了通知を受信 (S676) した  
登録装置は、次に、ステップS701において、共通鍵  
データの更新をサポートするデバイスとするか否かを判  
定する。デバイスに格納されたデータ中、そのいくつか  
は更新対象データとして前述したデータアップデートチ  
ケット (DUT : Data Update Ticket) (図32参照)  
を用いて更新が可能である。更新対象となるデータは、  
先に図33を用いて説明した通りである。このデータア  
ップデートチケット (DUT : Data Update Ticket) を  
用いた更新処理においても共通鍵方式、または公開鍵方  
式のいずれかの方式が可能であり、パーティションマネ  
ージャは設定したパーティションに応じていずれかの方  
式または両方式を実行可能なデータをデバイスを設定す  
る。

【0559】パーティションマネージャは、設定したパ  
ーティションを共通鍵方式によるデータ更新を実行する  
構成とする場合は、共通鍵方式のデータ更新処理に必要  
な情報 (ex. データアップデートチケット (DUT)  
のMAC検証用鍵他) をデバイスのパーティション鍵領  
域にセットし、デバイスが共通鍵認証を実行しないデバ  
イスであれば、これらの情報をデバイスのパーティシ  
ョン鍵領域に格納しない処理をする。

【0560】図64に示すように、データアップデート  
チケット (DUT : Data Update Ticket) を用いたデー

タ更新処理に共通鍵方式を用いる場合、ステップS70  
2~703、S711~S714を実行し、データ更新  
に共通鍵方式を用いない場合、これらのステップは省略  
される。

【0561】データ更新に共通鍵を用いる場合、ステ  
ップS702において登録装置は、データアップデートチ  
ケット (DUT : Data Update Ticket) 検証共通鍵書き  
込みコマンドとして、Kdut\_PAR1 :データアップデート  
チケット (DUT) のMAC検証用鍵、Kdut\_PAR2 :デー  
タ更新用暗号鍵、Kdut\_PAR3 :データアップデートチ  
ケット (DUT) のMAC検証用鍵、Kdut\_PAR4 :デー  
タ更新用暗号鍵およびこれらのバージョン情報をデバイ  
スに送信する。

【0562】ステップS711でデバイスは、上述の書  
き込みコマンドを受信し、ステップS712において、  
受領データをパーティション鍵領域 (図23参照) に書  
き込む。次にデータ書き込みによって生じたポインタ、  
サイズ、デバイス内のフリーブロック数の調整を実行  
(S713) し、書き込み終了通知を登録装置に送信  
(S714) する。

【0563】書き込み終了通知を受信 (S703) した  
登録装置は、ステップS704において、デバイスに設  
定したパーティションが公開鍵方式を用いたデータアッ  
プデートチケット (DUT : Data Update Ticket) を使  
用したデータ更新処理をサポートするか否かを判定す  
る。図64に示すように、公開鍵方式をサポートする場  
合、ステップS705~706、S715~S718を  
実行し、公開鍵方式をサポートしない場合、これらのス  
テップは省略される。

【0564】公開鍵方式をサポートする場合、ステップ  
S705において登録装置は、データアップデートチケ  
ット (DUT : Data Update Ticket) 発行者コード書き  
込みコマンドとして、DUTIC\_PAR (DUT Issuer Categor  
y) :データアップデートチケット (DUT : Data Updat  
e Ticket) 発行者カテゴリ、およびバージョン情報をデ  
バイスに送信する。

【0565】ステップS715でデバイスは、上述の書  
き込みコマンドを受信し、ステップS716において、  
受領データを公開鍵系パーティション鍵定義ブロック  
(PKDB (PUB) : Partition Key Definition Blo  
ck (PUB)) に書き込む。次にデータ書き込みによって生  
じたポインタ、サイズ、デバイス内のフリーブロック数  
の調整を実行 (S717) し、書き込み終了通知を登録  
装置に送信 (S718) し、登録装置が書き込み終了通  
知を受信 (S706) して処理を終了する。

【0566】パーティションマネージャによる初期登録  
処理 (図62~図64の処理フロー) が完了した状態の  
デバイスのメモリ内格納データ構成例を図65に示す。  
図65に示すパーティション (Partition) 領域中、パー  
ティション鍵領域には、上記のフロー (図62~図6



4)において、登録装置から送信されて下記のデータが書き込まれる。

- \* IRL\_PAR :パーティションアクセス排除デバイス (Device)、排除機器 (デバイスアクセス機器としてのリーダライタ、PC等のチケットユーザ、チケット発行手段)の識別子 (ID)を登録したリボケーションリスト (Revocation List (Device ID))
- \* CRL\_PAR :パーティションアクセス排除デバイス (Device)、排除機器 (デバイスアクセス機器としてのリーダライタ、PC等のチケットユーザ、チケット発行手段)の公開鍵証明書識別子 (ex. シリアルナンバ: SN)を登録したリボケーションリスト (Revocation List (Certificate))
- \* Kauth\_PAR\_B :双方向個別鍵認証用共通鍵
- \* Mkauth\_PAR\_A :双方向個別鍵認証用マスター鍵
- \* Kdut\_PAR1 :データアップデートチケット (DUT)のMAC検証用鍵
- \* Kdut\_PAR2 :データ更新用暗号鍵
- \* Kdut\_PAR3 :データアップデートチケット (DUT)のMAC検証用鍵
- \* Kdut\_PAR4 :データ更新用暗号鍵
- \* Kfrt :ファイル登録チケット (FRT)のMAC検証用鍵

【0567】また、

- \* PUB\_PAR :パーティション (Partition)の公開鍵
- \* PRI\_PAR :パーティション (Partition)の秘密鍵が、デバイスにおいて生成されて書き込まれる。

【0568】また、

- \* PARAM\_PAR :パーティション (Partition)の公開鍵パラメータ
- \* PUB\_CA(PAR) :認証局CA (PAR)の公開鍵  
共通鍵系パーティション鍵情報ブロック (Partition Key Definition Block(Common))  
公開鍵系パーティション鍵情報ブロック (Partition Key Definition Block(PUB))  
パーティション管理情報ブロック (Partition Management Information Block)

の各データは、パーティションの生成時 (処理フロー図60、図61参照)に書き込まれるデータである。

【0569】[B4. 2. パーティションマネージャ管理下における公開鍵証明書発行処理]次に図66以下を用いて、パーティションマネージャによるパーティション対応公開鍵証明書の発行処理について説明する。デバイスには、デバイス全体の認証、デバイスを単位とした処理に適用可能なデバイス対応公開鍵証明書 (CERT DEV)と、デバイス内の特定のパーティションに対する処理の際の認証その他検証処理等に適用可能なパーティション対応公開鍵証明書 (CERT PAR)が格納され得る。パーティション対応公開鍵証明書 (CERT PAR)は、デバイスに設定されたパーティション

毎に設定格納可能である。

【0570】パーティション対応公開鍵証明書 (CERT PAR)は、パーティションマネージャの管轄する登録局を介して認証局 (CA for PM) (図2、図3参照)の発行した公開鍵証明書をデバイスに付与する手続きにより発行され、登録局はパーティションマネージャの管轄登録局が発行した公開鍵証明書 (CERT PAR)についての管理 (データベース332 (図9参照))を実行する。

【0571】図66および図67に従って、パーティションマネージャの管轄登録局による設定パーティションに対するパーティション対応公開鍵証明書 (CERT PAR)の発行処理の手順を説明する。図66、図67において、左側がパーティションマネージャの管轄登録局のCERT (公開鍵証明書)発行装置、具体的には、図9に示すパーティションマネージャの構成図における制御手段331の処理、右側がデバイスの処理である。

【0572】まずステップS721において、CERT発行装置は、パーティション対応公開鍵証明書 (CERT PAR)の発行対象となるデバイスのユーザ情報を取得し、証明書発行の許可 (判定)を行ない発行対象となるデバイスとの通信路を確保する。パーティション対応公開鍵証明書 (CERT PAR)の発行対象となるデバイスのユーザ情報は、例えばデバイスの初期登録時に生成したデータから取得可能である。なお、ユーザ情報はデバイスとの通信路設定後、デバイスから取得してもよい。通信路は、有線、無線を問わずデータ送受信可能な通信路として確保されればよい。

【0573】次にCERT発行装置は、ステップS722において、乱数Rを含む認証データ生成コマンドをデバイスに対して送信する。認証データ生成コマンドを受信 (S731)したデバイスは、受信乱数Rと、デバイス識別子 (IDm)の結合データにデバイス秘密鍵 (PRI PAR)を適用してデジタル署名 (S)の生成処理 (図12参照)を実行 (S732)する。デバイスは、デバイスの識別データ (IDm)と署名 (S)をCERT発行装置に送信する。

【0574】デバイスから識別データ (IDm)と署名 (S)を受信 (S723)したCERT発行装置は、受信したデバイス識別データ (IDm)を検索キーとしてデータベースDB (PAR)332から格納済みのデバイス公開鍵 (PUB PAR)を取得する。さらに、取得したデバイス公開鍵 (PUB PAR)を適用して署名 (S)の検証処理 (図13参照)を実行 (S725)する。検証に成功しなかった場合は、デバイスからの送信データは不正なデータであると判定し処理は終了する。

【0575】検証に成功した場合は、認証局 (CA for PM)620に対してパーティション対応公開鍵証明書 (CERT PAR)の発行処理を依頼 (S72

7) する。パーティションマネージャは認証局 620 の発行したパーティション対応公開鍵証明書 (CERT PAR) を受信 (S728) してデバイスに送信 (S729) する。

【0576】パーティションマネージャ (登録局) からパーティション対応公開鍵証明書 (CERT PAR) を受信したデバイスは、予めパーティション鍵領域 (図23参照) に格納済みの認証局の公開鍵 (PUB CA (PAR)) を用いて受信したパーティション対応公開鍵証明書 (CERT PAR) の署名検証を実行する。すなわち公開鍵証明書には認証局の秘密鍵で実行され署名があり (図11参照)、この署名検証 (S735) を行なう。

【0577】署名検証に失敗した場合は、正当な公開鍵証明書でないと判定し、エラー通知を CERT 発行装置に対して実行 (S745) する。

【0578】署名検証に成功した場合は、パーティション対応公開鍵証明書 (CERT PAR) に格納されたデバイス公開鍵 (PUB PAR) と自デバイスに保管されたデバイス公開鍵 (PUB PAR) の比較を実行 (S741) し、一致しない場合はエラー通知を実行し、一致した場合は、受信したパーティション対応公開鍵証明書 (CERT PAR) をパーティション鍵領域 (図23参照) に格納 (S743) する。なお、パーティション対応公開鍵証明書 (CERT PAR) の発行以前は、この領域に自デバイスで生成した公開鍵 (PUB PAR) を格納し、正当なパーティション対応公開鍵証明書 (CERT PAR) が発行された時点で、パーティション対応公開鍵証明書 (CERT PAR) により上書きする処理として格納する。

【0579】パーティション対応公開鍵証明書 (CERT PAR) の格納が終了すると格納処理終了通知を CERT 発行装置に送信 (S744) する。CERT 発行装置は、格納処理終了通知を受信 (S751) し、格納成功を確認 (S752) して処理を終了する。格納成功の確認が得られない場合はエラーとして処理が終了する。

【0580】[B4. 3. パーティション生成処理各方式における処理手順] 上述したように、パーティションの設定登録処理において、パーティションマネージャの管理するデバイスアクセス機器としてのリーダーとデバイス間において、相互認証が実行され、パーティション登録チケット (PRT) に基づくパーティションの設定がなされる。上述したように相互認証処理の様態は、公開鍵相互認証、共通鍵相互認証の2種類のいずれかであり、またチケット (PRT) の検証処理も公開鍵系の署名検証、共通鍵系の MAC 検証の2種類のいずれかが実行されることになる。すなわち処理態様としては大きく分けて、

(A) 相互認証 (公開鍵)、チケット (PRT) 検証

(公開鍵)

(B) 相互認証 (公開鍵)、チケット (PRT) 検証 (共通鍵)

(C) 相互認証 (共通鍵)、チケット (PRT) 検証 (共通鍵)

(D) 相互認証 (共通鍵)、チケット (PRT) 検証 (公開鍵)

の4態様がある。

【0581】これらの4態様についての処理を、認証局 (CA (DM))、デバイスマネージャ (DM)、パーティションマネージャ (PM)、デバイス、各エンティティ間において実行されるデータ転送処理を中心として図を用いて簡潔に説明する。

【0582】(A) 相互認証 (公開鍵)、チケット (PRT) 検証 (公開鍵)

まず、相互認証処理に公開鍵方式を適用し、チケット (PRT) 検証に公開鍵方式を適用する場合の各エンティティ間のデータ転送について図68を用いて説明する。なお以下では、説明を簡略化するために図68に示すように、認証局 (CA) を1つとし、登録局をデバイスマネージャ内に1つ設定し、デバイスマネージャ公開鍵証明書 (Cert. DM)、パーティションマネージャ公開鍵証明書 (Cert. PM) の双方をこれらの各登録局、認証局を介して発行する構成とした。またパーティション登録チケット (PRT) の発行手段はデバイスマネージャ (DM) であり、パーティション登録チケット (PRT) に対する署名はデバイスマネージャの秘密鍵を用いて実行される。

【0583】図に示す番号順に各エンティティ間でデータ転送が実行される。以下、各番号に従って処理を説明する。

(1) デバイスマネージャ (DM) の公開鍵証明書 (Cert. DM) の発行、公開鍵証明書 (Cert. DM) は、認証局 (CA) によってデバイスマネージャの発行要求により、登録局を介した証明書発行手続きによってデバイスマネージャに対して発行される。

(2) パーティションマネージャ (PM) の公開鍵証明書 (Cert. PM) の発行、公開鍵証明書 (Cert. PM) は、認証局 (CA) によってパーティションマネージャの発行要求により、登録局を介した証明書発行手続きによってパーティションマネージャに対して発行される。

【0584】(3) パーティション登録チケット (PRT) の発行処理

パーティション登録チケット (PRT) は、デバイスマネージャの管理するパーティション登録チケット発行手段 (PRT Ticket Issuer) によりパーティションマネージャ (PM) に対して発行される。この場合、公開鍵方式の署名生成、検証を実行するため、デバイスマネージャの秘密鍵による署名 (Signature) が生成 (図12参

照)されてPRTに付加される。

(4) PRTおよびDM公開鍵証明書(Cert. DM)のPMに対する供給デバイスマネージャの管理するパーティション登録チケット発行手段(PRTTicket Issuer)により発行されたパーティション登録チケット

(PRT)は、DM公開鍵証明書(Cert. DM)とともにパーティションマネージャに対して送信される。

【0585】(5) PMとデバイス間の相互認証発行されたPRTに従ったパーティションを生成しようとする対象のデバイスと、パーティションマネージャ(具体的にはチケットユーザであるデバイスアクセス機器としてのリーダライタ)は、公開鍵方式の相互認証(図50参照)を実行する。

【0586】(6) PRTおよびDM公開鍵証明書(Cert. DM)のデバイスに対する供給PMとデバイス間の相互認証が成立すると、パーティションマネージャ(PM)は、デバイスに対してパーティション登録チケット(PRT)、およびDM公開鍵証明書(Cert. DM)を送信する。デバイスは、受信したパーティション登録チケット(PRT)について、

(1) チケット発行者(Ticket Issuer)=DMの公開鍵証明書(CERT)が改竄されたものでない正当な公開鍵証明書(CERT)であること、(2) チケット発行者(Ticket Issuer)の公開鍵証明書(CERT)のオプション領域に記録されたコードと、デバイス内のDKDB(Device Key Definition Block)(PUB)に記録されたチケット発行手段コード(PRTIC:PRT Issuer Code)の一致、(3) チケット発行手段(Ticket Issuer)がリボークされていないこと、(4) 受信チケット(PRT)の署名(Signature)の検証によりチケットが改竄のないことの確認を実行し、さらに、PRTチケットに格納されたPRTユーザ(この場合はPM:チケットユーザであるデバイスアクセス機器としてのリーダライタ)と受信したパーティションマネージャの公開鍵証明書の識別データ(DN)として記録された識別子またはカテゴリまたはシリアル(SN)の一致を確認し、相互認証済みであることを確認することによりPRTユーザ(PM:デバイスアクセス機器としてのリーダライタ)の検証(図57、図58参照)を実行する。

【0587】(7) パーティションの生成  
パーティション登録チケット(PRT)の検証、PRT発行者(PRT Issuer)、PRTユーザの検証に成功すると、パーティション登録チケット(PRT)に記述されたルールに従ってパーティションがデバイスのメモリ部に生成(図60、図61参照)される。

【0588】(8) 鍵データ書き込み  
パーティションがデバイスのメモリ部に生成されると、生成されたパーティション内に対する各種鍵の格納処理が実行される。(9) 公開鍵の読み出し、

(10) 公開鍵証明書の発行

生成パーティションに対する各種サービス時の認証処理(パーティション生成、ファイル生成、ファイルアクセス、データアップデート等のサービス利用時の認証処理)に際し、公開鍵認証を行なう場合、デバイスは公開鍵、秘密鍵の鍵ペアを生成し、生成した公開鍵をパーティションマネージャに送信し、登録局、認証局を介して公開鍵証明書の発行処理を行ない、発行された公開鍵証明書をパーティション鍵領域(図23参照)に格納する。この際、生成した公開鍵の格納領域に対して発行された公開鍵証明書を格納する。なお、この(9)、(10)の処理は、作成パーティションに対する各種サービス時の認証処理(パーティション生成、ファイル生成、ファイルアクセス、データアップデート等のサービス利用時の認証処理)の際に公開鍵認証を行なう構成の場合に実行すればよい。

【0589】以上の処理によって、相互認証(公開鍵)、チケット(PRT)検証(公開鍵)の各方式に従ったパーティションの生成処理が実行される。

【0590】(B) 相互認証(公開鍵)、チケット(PRT)検証(共通鍵)

次に、相互認証処理に公開鍵方式を適用し、チケット(PRT)検証に共通鍵方式を適用する場合の各エンティティ間のデータ転送について図69を用いて説明する。図に示す番号順に各エンティティ間でデータ転送が実行される。以下、各番号に従って処理を説明する。

【0591】(1) パーティションマネージャ(PM)の公開鍵証明書(Cert. PM)の発行、公開鍵証明書(Cert. PM)は、認証局(CA)によってパーティションマネージャの発行要求により、登録局を介した証明書発行手続きによってデバイスマネージャに対して発行される。

【0592】(2) パーティション登録チケット(PRT)の発行処理

パーティション登録チケット(PRT)は、デバイスマネージャの管理するパーティション登録チケット発行手段(PRT Ticket Issuer)によりパーティションマネージャ(PM)に対して発行される。この場合、共通鍵方式の検証値としてMAC(Message Authentication Code)(図59参照)がPRTに付加される。

(3) PRTのPMに対する供給  
デバイスマネージャの管理するパーティション登録チケット発行手段(PRTTicket Issuer)により発行されたパーティション登録チケット(PRT)は、パーティションマネージャに対して送信される。

【0593】(4) PMとデバイス間の相互認証  
発行されたPRTに従ったパーティションを生成しようとする対象のデバイスと、パーティションマネージャ(具体的にはチケットユーザであるデバイスアクセス機器としてのリーダライタ)は、公開鍵方式の相互認証(図50参照)を実行する。

#### 【0594】(5) PRTの送信

パーティションマネージャは発行されたパーティション登録チケット(PRT)をデバイスに送付する。デバイスは、受信したパーティション登録チケット(PRT)についてMAC検証処理を実行し、PRT発行者(PRT Issuer)の検証、さらに、PRTチケットに格納されたPRTユーザ(この場合はPM:チケットユーザであるデバイスアクセス機器としてのリーダライタ)と受信したパーティションマネージャの公開鍵証明書の識別データ(DN)として記録された識別子またはカテゴリまたはシリアル(SN)の一致を確認し相互認証済みであることを確認することによりPRTユーザ(PM:デバイスアクセス機器としてのリーダライタ)の検証(図57、図58参照)を実行する。

#### 【0595】(6) パーティションの生成

パーティション登録チケット(PRT)の検証、PRT発行者(PRT Issuer)、PRTユーザの検証に成功すると、パーティション登録チケット(PRT)に記述されたルールに従ってパーティションがデバイスのメモリ部に生成(図60、図61参照)される。

#### (7) 鍵データ書き込み

パーティションがデバイスのメモリ部に生成されると、生成されたパーティション内に対する各種鍵の格納処理が実行される。

#### 【0596】(8) 公開鍵の読み出し、

#### (9) 公開鍵証明書の発行

生成パーティションに対する各種サービス時の認証処理(パーティション生成、ファイル生成、ファイルアクセス、データアップデート等のサービス利用時の認証処理)に際し、公開鍵認証を行なう場合、デバイスは公開鍵、秘密鍵の鍵ペアを生成し、生成した公開鍵をパーティションマネージャに送信し、登録局、認証局を介して公開鍵証明書の発行処理を行ない、発行された公開鍵証明書をパーティション鍵領域(図23参照)に格納する。この際、生成した公開鍵の格納領域に対して発行された公開鍵証明書を格納する。なお、この(8)、

(9)の処理は、作成パーティションに対する各種サービス時の認証処理(パーティション生成、ファイル生成、ファイルアクセス、データアップデート等のサービス利用時の認証処理)の際に公開鍵認証を行なう構成の場合に実行すればよい。

【0597】以上の処理によって、相互認証(公開鍵)、チケット(PRT)検証(共通鍵)の各方式に従ったパーティションの生成処理が実行される。

【0598】(C) 相互認証(共通鍵)、チケット(PRT)検証(共通鍵)

次に、相互認証処理に共通鍵方式を適用し、チケット(PRT)検証に共通鍵方式を適用する場合の各エンティティ間のデータ転送について図70を用いて説明する。図に示す番号順に各エンティティ間でデータ転送が

実行される。以下、各番号に従って処理を説明する。

【0599】(1) パーティション登録チケット(PRT)の発行処理

パーティション登録チケット(PRT)は、デバイスマネージャの管理するパーティション登録チケット発行手段(PRT Ticket Issuer)によりパーティションマネージャ(PM)に対して発行される。この場合、共通鍵方式の検証値としてMAC(図59参照)がPRTに付加される。

【0600】(2) PRTのPMに対する供給

デバイスマネージャの管理するパーティション登録チケット発行手段(PRT Ticket Issuer)により発行されたパーティション登録チケット(PRT)は、パーティションマネージャに対して送信される。

【0601】(3) PMとデバイス間の相互認証

発行されたPRTに従ったパーティションを生成しようとする対象のデバイスと、パーティションマネージャ(具体的にはチケットユーザであるデバイスアクセス機器としてのリーダライタ)は、共通鍵方式の相互認証(図53、図54参照)を実行する。

【0602】(4) PRTの送信

パーティションマネージャは発行されたパーティション登録チケット(PRT)をデバイスに送付する。デバイスは、受信したパーティション登録チケット(PRT)についてMAC検証処理を実行し、PRT発行者(PRT Issuer)の検証、さらに、PRTチケットに格納されたPRTユーザ(この場合はPM:チケットユーザであるデバイスアクセス機器としてのリーダライタ)とパーティションマネージャの識別子の一致を確認し、相互認証済みであることを確認することによりPRTユーザ(PM:デバイスアクセス機器としてのリーダライタ)の検証(図57、図58参照)を実行する。

【0603】(5) パーティションの生成

パーティション登録チケット(PRT)の検証、PRT発行者(PRT Issuer)、PRTユーザの検証に成功すると、パーティション登録チケット(PRT)に記述されたルールに従ってパーティションがデバイスのメモリ部に生成(図60、図61参照)される。

【0604】(6) 鍵データ書き込み

パーティションがデバイスのメモリ部に生成されると、生成されたパーティション内に対する各種鍵の格納処理が実行される。

(7) 公開鍵の読み出し、

(8) 公開鍵証明書の発行

生成パーティションに対する各種サービス時の認証処理(パーティション生成、ファイル生成、ファイルアクセス、データアップデート等のサービス利用時の認証処理)に際し、公開鍵認証を行なう場合、デバイスは公開鍵、秘密鍵の鍵ペアを生成し、生成した公開鍵をパーティションマネージャに送信し、登録局、認証局を介して

公開鍵証明書の発行処理を行ない、発行された公開鍵証明書をパーティション鍵領域(図23参照)に格納する。この際、生成した公開鍵の格納領域に対して発行された公開鍵証明書を格納する。なお、この(7)、

(8)の処理は、作成パーティションに対する各種サービス時の認証処理(パーティション生成、ファイル生成、ファイルアクセス、データアップデート等のサービス利用時の認証処理)の際に公開鍵認証を行なう構成の場合に実行すればよい。

【0605】以上の処理によって、相互認証(共通鍵)、チケット(PRT)検証(公開鍵)の各方式に従ったパーティションの生成処理が実行される。

【0606】(D)相互認証(共通鍵)、チケット(PRT)検証(公開鍵)

次に、相互認証処理に共通鍵方式を適用し、チケット(PRT)検証に公開鍵方式を適用する場合の各エンティティ間のデータ転送について図71を用いて説明する。図に示す番号順に各エンティティ間でデータ転送が実行される。以下、各番号に従って処理を説明する。

【0607】(1)デバイスマネージャ(DM)の公開鍵証明書(Cert. DM)の発行、公開鍵証明書(Cert. DM)は、認証局(CA)によってデバイスマネージャの発行要求により、登録局を介した証明書発行手続きによってデバイスマネージャに対して発行される。

【0608】(2)パーティション登録チケット(PRT)の発行処理

パーティション登録チケット(PRT)は、デバイスマネージャの管理するパーティション登録チケット発行手段(PRT Ticket Issuer)によりパーティションマネージャ(PM)に対して発行される。この場合、公開鍵方式の署名生成、検証を実行するため、デバイスマネージャの秘密鍵による署名(Signature)が生成(図12参照)されてPRTに付加される。

(3)PRTおよびDM公開鍵証明書(Cert. DM)のPMに対する供給デバイスマネージャの管理するパーティション登録チケット発行手段(PRT Ticket Issuer)により発行されたパーティション登録チケット

(PRT)は、DM公開鍵証明書(Cert. DM)とともにパーティションマネージャに対して送信される。

【0609】(4)PMとデバイス間の相互認証  
発行されたPRTに従ったパーティションを生成しようとする対象のデバイスと、パーティションマネージャ(具体的にはチケットユーザであるデバイスアクセス機器としてのリーダーライタ)は、共通鍵方式の相互認証(図53、図54参照)を実行する。

【0610】(5)PRTおよびDM公開鍵証明書(Cert. DM)のデバイスに対する供給  
PMとデバイス間の相互認証が成立すると、パーティションマネージャ(PM)は、デバイスに対してパーティ

ション登録チケット(PRT)、およびDM公開鍵証明書(Cert. DM)を送信する。デバイスは、受信したパーティション登録チケット(PRT)について、

(1)チケット発行者(Ticket Issuer)=DMの公開鍵証明書(CERT)が改竄されたものでない正当な公開鍵証明書(CERT)であること、(2)チケット発行者(Ticket Issuer)の公開鍵証明書(CERT)のオプション領域に記録されたコードと、デバイス内のDKDB(Device Key Definition Block)(PUB)に記録されたチケット発行手段コード(PRTIC:PRT Issuer Code)の一致、(3)チケット発行手段(Ticket Issuer)がリボークされていないこと、(4)受信チケット(PRT)の署名(Signature)の検証によりチケットが改竄のないことの確認を実行し、さらに、PRTチケットに格納されたPRTユーザ(この場合はPM:チケットユーザであるデバイスアクセス機器としてのリーダーライタ)とパーティションマネージャの公開鍵証明書中の識別データ(DN)として記録された識別子またはカテゴリまたはシリアル(SN)の一致を確認し、相互認証済みであることを確認することによりPRTユーザ(PM:デバイスアクセス機器としてのリーダーライタ)の検証(図57、図58参照)を実行する。

【0611】(6)パーティションの生成  
パーティション登録チケット(PRT)の検証、PRT発行者(PRT Issuer)、PRTユーザの検証に成功すると、パーティション登録チケット(PRT)に記述されたルールに従ってパーティションがデバイスのメモリ部に生成(図60、図61参照)される。

【0612】(7)鍵データ書き込み  
パーティションがデバイスのメモリ部に生成されると、生成されたパーティション内に対する各種鍵の格納処理が実行される。

(8)公開鍵の読み出し、

(9)公開鍵証明書の発行

生成パーティションに対する各種サービス時の認証処理(パーティション生成、ファイル生成、ファイルアクセス、データアップデート等のサービス利用時の認証処理)に際し、公開鍵認証を行なう場合、デバイスは公開鍵、秘密鍵の鍵ペアを生成し、生成した公開鍵をパーティションマネージャに送信し、登録局、認証局を介して公開鍵証明書の発行処理を行ない、発行された公開鍵証明書をパーティション鍵領域(図23参照)に格納する。この際、生成した公開鍵の格納領域に対して発行された公開鍵証明書を格納する。なお、この(8)、

(9)の処理は、作成パーティションに対する各種サービス時の認証処理(パーティション生成、ファイル生成、ファイルアクセス、データアップデート等のサービス利用時の認証処理)の際に公開鍵認証を行なう構成の場合に実行すればよい。

【0613】以上の処理によって、相互認証(共通

鍵)、チケット(PRT)検証(公開鍵)の各方式に従ったパーティションの生成処理が実行される。

【0614】[B4. 4. ファイル登録チケット(FRT)を利用したファイル生成、削除処理]次に、デバイスに生成したパーティション内にファイル登録チケット(FRT)を適用してファイルを生成、または削除する処理について説明する。図72以下のフロー他の図面を参照して説明する。なお、ファイル作成、削除処理には、デバイスとデバイスアクセス機器としてのリーダライタ(パーティションマネージャ)間における相互認証処理(デバイス認証またはパーティション認証)、パーティション登録チケット(FRT:File Registration Ticket)の正当性検証処理が含まれる。

【0615】図72に示すファイル生成、削除処理フローについて説明する。図72において、左側がパーティションマネージャのファイル作成・削除装置、右側がデバイス(図5参照)の処理を示す。なお、パーティションマネージャのファイル作成・削除装置は、デバイスに対するデータ読み取り書き込み処理可能な装置(ex. デバイスアクセス機器としてのリーダライタ、PC)であり、図10のデバイスアクセス機器としてのリーダライタに相当する。まず、図72を用いて、ファイル作成、削除処理の概要を説明し、その後、当処理に含まれるファイル作成、削除操作の詳細を図73のフローを用いて説明する。

【0616】まず、図72のステップS801とS810において、ファイル作成・削除装置とデバイス間での相互認証処理が実行される。データ送受信を実行する2つの手段間では、相互に相手が正しいデータ通信者であるか否かを確認して、その後に必要なデータ転送を行なうことが行われる。相手が正しいデータ通信者であるか否かの確認処理が相互認証処理である。相互認証処理時にセッション鍵の生成を実行して、生成したセッション鍵を共有鍵として暗号化処理を実行してデータ送信を行なう。

【0617】相互認証処理については、先のパーティション生成、削除処理の欄で説明したと同様の処理であり、パーティション認証が実行される。それぞれについて共通鍵方式認証、あるいは公開鍵方式認証処理のいずれかが適用される。この相互認証処理は、前述の図48～図56を用いて説明したと同様の処理であるので説明を省略する。

【0618】なお、相互認証処理として実行すべき処理は、適用するファイル登録チケット(FRT)(図27参照)の

\* Authentication Flag:チケット(Ticket)の利用処理においてデバイス(Device)との相互認証が必要か否かを示すフラグ

\* Authentication Type:デバイス(Device)の相互認証のタイプ(公開鍵認証、または、共通鍵認証、また

は、いずれでも可(Any))によって決定される。

【0619】認証処理に失敗した場合(S802, S811でNo)は、相互が正当な機器、デバイスであることの確認がとれないことを示し、以下の処理は実行されずエラーとして処理は終了する。

【0620】認証処理に成功すると、ファイル作成・削除装置は、デバイスに対してファイル登録チケット(FRT:File Registration Ticket)を送信する。ファイル登録チケット(FRT)は、パーティションマネージャの管理下のファイル登録チケット(FRT)発行手段(FRT Issuer)により発行されるチケットである。ファイル登録チケット(FRT)は、デバイスに対するアクセス制御チケットであり、先に説明した図27のデータフォーマット構成を持つチケットである。

【0621】なお、ファイル登録チケット(FRT)を、チケットユーザに対して送信する際には、公開鍵方式の場合、ファイル登録チケット(FRT)発行手段(FRT Issuer)の公開鍵証明書(CERT\_FRTI)も一緒に送信する。FRT発行手段の公開鍵証明書(CERT\_FRTI)の属性(Attribute)は、ファイル登録チケット(FRT)発行手段(FRT Issuer)の識別子(FRTIC)と一致する。

【0622】ファイル登録チケット(FRT)を受信(S812)したデバイスは、受信したチケット(FRT)の正当性と利用者チェック処理を実行(S813)する。チケットの正当性の検証処理は、共通鍵方式によるMAC検証、あるいは公開鍵方式による署名検証処理のいずれかを適用して実行される。利用者チェックは、チケットを送信してきた機器(チケット利用者)の正当性をチェックする処理であり、相互認証が成立済みであり、認証相手の識別データと、チケットに記録されているチケットユーザ識別子(図27参照)との一致等を検証する処理として実行される。これらの処理は、先のパーティション登録チケット(PRT)の適用処理についての説明中、図57～図59を用いて説明したと同様の処理であるので説明を省略する。

【0623】デバイスにおいて、受信チケット(FRT)の正当性と利用者チェック処理の結果、チケットおよび利用者の正当なことが確認できなかった場合(S814でNo)は、ファイル登録チケット(FRT)受理エラーをファイル作成・削除装置に通知(S818)する。チケットおよび利用者の正当なことが確認できた場合(S814でYes)は、受信したファイル登録チケット(FRT)に記述されたルールに従いデバイス内のメモリ部におけるファイルの生成、または削除処理を実行する。この処理の詳細については、別フローを用いて後段で詳述する。

【0624】ファイル登録チケット(FRT)の記述に従って、ファイルの生成または削除処理に成功(S816でYes)すると、FRT受理成功をファイル作成・

削除装置に通知 (S 8 1 7) する。一方、ファイルの生成または削除処理に失敗 (S 8 1 6 で N o) した場合は、F R T 受理エラーをファイル作成・削除装置に通知 (S 8 1 8) する。

【0 6 2 5】ファイル作成・削除装置は、F R T 受理結果を受信 (S 8 0 4) し、F R T 処理結果を判定し、F R T 受理結果がエラーである場合 (S 8 0 5 で N o) は、エラーとして処理を終了し、F R T 受理結果が成功 (S 8 0 5 で Y e s) である場合はセッションクリアコマンドの送受信 (S 8 0 6, S 8 1 9) を実行し、デバイス側に生成した認証テーブルを破棄 (S 8 2 0) し、処理を終了する。認証テーブルは、ステップ S 8 0 1, S 8 1 0 の相互認証処理において生成されるテーブルであり、前述したパーティション登録チケット (P R T) の適用処理の項目において説明した構成、すなわち、図 5 1 の構成と同様のものである。

【0 6 2 6】このようにファイル登録チケット (F R T) を利用して、デバイス内に設定されたパーティション内にファイルの生成、または生成済みのファイルの削除処理が実行される。以下、当処理に含まれるファイルの生成、削除処理 (S 8 1 5) について、図 7 3 を用いて説明する。

【0 6 2 7】(ファイル作成・削除処理) 図 7 2 のフローに示すステップ S 8 1 5 において実行されるファイル登録チケット (F R T) に基づくパーティションの作成、削除処理の詳細について、図 7 3 の処理フローを用いて説明する。ファイルの作成、削除処理は、チケットユーザ (e x. デバイスアクセス機器としてのリーダライタ、P C 等) からファイル登録チケット (F R T) を受信したデバイスが、ファイル登録チケット (F R T) に基づいて実行する処理である。

【0 6 2 8】図 7 3 のステップ S 8 2 1 において、デバイスは、受信したファイル登録チケット (F R T: File Registration ticket) に記録された処理タイプ、すなわち Operation Type (パーティション (Partition) 作成か削除かの指定 (作成 (Generate) / 削除 (Delete))) を検証する。処理タイプ (Operation Type) が、ファイル作成である場合、ステップ S 8 2 2 以下を実行し、ファイル削除である場合、ステップ S 8 4 1 以下を実行する。

【0 6 2 9】まず、ファイル作成処理について説明する。デバイスはステップ S 8 2 2 において、ファイル登録チケット (F R T) に記述されたファイル識別子 (I D) と同一 I D のファイルがデバイスの処理対象パーティション内に存在するか否かを検証する。この判定は、デバイスのメモリ部に設定されたパーティション領域のファイル定義ブロック (図 2 4 参照) に受信チケット (F R T) に記述されたファイル I D と同一のファイル I D が記述されているか否かを検証することによって判定可能である。

【0 6 3 0】すでにデバイスに同一 I D のファイルが存在する場合は、同一 I D を持つ重複ファイルを同一のパーティション内に存在させることは許されないため、ファイルの生成は実行せず、エラー終了とする。同一 I D のファイルが処理対象パーティション内に存在しない場合は、ステップ S 8 2 3 において、パーティション管理情報ブロック (図 2 0 参照) のパーティション内の空きブロック数 (Free Block Number in Partition) と、ファイル登録チケット (F R T) に記述されたファイルサイズ (File Size) とを比較し、チケット (F R T) に記述されたファイルサイズ (File Size) 以上の空きブロック領域がデバイスの処理対象パーティション内に存在するか否かを判定する。存在しない場合は、F R T に記述されたサイズのファイルの生成はできないため、エラー終了とする。

【0 6 3 1】チケット (F R T) に記述されたファイルサイズ (File Size) 以上の空きブロック領域がデバイスのメモリ部の処理対象パーティション内に存在すると判定された場合は、ステップ S 8 2 4 に進み、パーティション管理情報ブロックの空き領域ポインタ (Pointer of Free Area) を参照してパーティションの空き領域 (Free Area in Partition) の最上位ブロックにファイル定義ブロック (F D B) エリア (図 2 4 参照) を確保する。

【0 6 3 2】次に、デバイスは、確保したファイル定義ブロック (F D B) エリアに、ファイル登録チケット (F R T) に記述されたファイル I D のコピーを実行 (S 8 2 5) し、さらに、ファイル定義ブロック (F D B) エリアのファイルスタート位置 (File Start Position) に、パーティション管理情報ブロック (図 2 0 参照) の空き領域ポインタ (Pointer of Free Area) のコピー処理を実行 (S 8 2 6) する。

【0 6 3 3】さらに、ステップ S 8 2 7 において、ファイル定義ブロック (F D B) のファイルサイズ (File Size)、サービス許可チケット発行手段コード (S P T I C)、およびバージョン、(S P T I C Version)、ファイル構造タイプ (File Structure Type Code)、ファイルアクセスを行う際に、指定する認証方式 (Acceptable Authentication Type)、指定する検証方式 (Acceptable Verification Type) のそれぞれに、ファイル登録チケット (F R T) に記述された各対応データをコピーする。

【0 6 3 4】次に、ステップ S 8 2 8 において、ファイル登録チケット (F R T) に格納された Kspt\_Encrypted (ファイル定義ブロック (File Definition Block) に記載されるサービス許可チケット (S P T) の MAC 検証用鍵 Kspt をそのパーティションのファイル登録チケットの MAC 検証用鍵 Kfrt で暗号化したデータ Kfrt (Kspt)) をファイル登録チケットの MAC 検証用鍵 Kfrt を用いて復号してファイル定義ブロック (F D B)

に格納する。なお、ファイル登録チケットのMAC検証用鍵 Kfrtは、パーティションの生成時にパーティション鍵領域に格納済みである。

【0635】次に、ステップS829において、パーティション管理情報ブロック（図20参照）のパーティション（Partition）内の空きブロック数（Free Block Number in Partition）からファイルサイズ（File Size）+1を減算する。なお、+1は、ファイル定義ブロック（FDB）用のブロックを意味する。

【0636】次に、ステップS830において、パーティション管理情報ブロック（図20参照）の空き領域ポインタ（Pointer of Free Area）に生成したファイルサイズ（File Size）を加算し、ステップS831において、パーティション管理情報ブロックのファイル数（File Number）に1を加算、すなわち生成したファイル数（1）を加算する。

【0637】次に、ステップS832において、ファイル登録チケット（FRT）に格納されたFile Structure（生成するファイル（File）のファイル構造（Structure））に応じた初期化処理を実行する。例えばファイル構造がランダム(Random)であれば、0リセット、サイクリック(Cyclic)であれば、ポインタ、データを0リセットするなどの処理を実行する。これらの処理により、生成したパーティション内に新たなファイルが生成される。

【0638】次に図73のステップS841～S848のファイル削除処理について説明する。ステップS841では、ファイル登録チケット（FRT）に記述されたファイルIDと同一IDのファイルがデバイスのメモリ部の処理対象パーティション内に存在するか否かを検証する。この判定は、デバイスのメモリ部のファイル定義ブロック（図24参照）に受信チケット（FRT）に記述されたファイルIDと同一のファイルIDが記述されているか否かを検証することによって判定可能である。

【0639】デバイスの処理対象パーティション内に同一ファイルIDのファイルが存在しない場合は、ファイルの削除は不可能であるので、エラー終了とする。同一IDのファイルがデバイスの処理対象パーティション内に存在する場合は、ステップS842において、削除対象のファイルより後に生成されたファイルが処理対象パーティション内に存在するか否かを判定する。存在しない場合は、削除対象のファイルが最新のファイルであり、ステップS849において削除対象のファイルのファイル定義ブロック（FDB）（図24参照）を削除する。

【0640】ステップS842において、削除対象のファイルより後に生成されたファイルが処理対象パーティション内に存在すると判定された場合は、後に生成されたファイル（後ファイル）のデータを削除対象のファイルのサイズ（FS）分、下位にずらす処理を実行（S8

43）し、さらに、後ファイルのファイル定義ブロック（FDB）を1ブロック上位にずらす処理を実行（S844）する。また、後ファイルのファイル定義ブロック（FDB）に記録されたファイル開始位置（File Start Portion）から削除ファイルのサイズ（FS）を減算する処理を実行する（S845）。

【0641】ステップS845またはS849の処理の後、ステップS846において、パーティション管理情報ブロック（PMIB）（図20参照）のパーティション内の空きブロック数（Free Block Number in Partition）に削除ファイルのサイズ（FS）+1を加算する。+1は、削除ファイルのファイル定義ブロック（FDB）用のブロックを意味する。

【0642】次にステップS847において、パーティション管理情報ブロック（PMIB）（図20参照）の空き領域ポインタ（Pointer of Free Area）の値から削除ファイルのサイズ（FS）を減算する。さらに、ステップS848において、パーティション管理情報ブロック（PMIB）（図20参照）のファイル数（File Number）から1を減算、すなわち削除したファイル数

（1）を減算してファイル登録チケット（FRT）に基づくファイル削除処理が終了する。

【0643】以上が、図72の処理フローにおけるステップS815のファイル登録チケット（FRT）に基づくファイル生成、削除処理である。

【0644】パーティションマネージャによるファイル生成処理が完了した状態のデバイスのメモリ内格納データ構成例を図74に示す。図74に示すパーティション（Partion）領域中、

ファイル定義ブロック（1～N）（File Definition Block）

パーティション鍵領域（Partition Key Area）

共通鍵系パーティション鍵情報ブロック（Partition Key Definition Block(Common)）

公開鍵系パーティション鍵情報ブロック（Partition Key Definition Block(PUB)）

パーティション管理情報ブロック（Partition Management Information Block）

の各データは、ファイル生成時、またはパーティション生成時に書き込まれるデータである。ファイル領域（File Data Area 1～N）は、ファイル生成処理によって処理対象パーティション内にファイル領域として確保される。

【0645】[B4. 5. ファイル生成処理各方式における処理手順] 上述したファイルの設定登録処理において、パーティションマネージャが管理し、ファイル登録チケットユーザであるデバイスアクセス機器としてのリーダーライタとデバイス間において、相互認証が実行され、ファイル登録チケット（FRT）に基づくファイルの設定がなされる。相互認証処理の態様は、公開鍵相互



認証、共通鍵相互認証の2種類のいずれかであり、またチケット (FRT) の検証処理も公開鍵系の署名検証、共通鍵系のMAC検証の2種類のいずれかが実行されることになる。すなわち処理態様としては大きく分けて、

- (A) 相互認証 (公開鍵)、チケット (FRT) 検証 (公開鍵)
- (B) 相互認証 (公開鍵)、チケット (FRT) 検証 (共通鍵)
- (C) 相互認証 (共通鍵)、チケット (FRT) 検証 (共通鍵)
- (D) 相互認証 (共通鍵)、チケット (FRT) 検証 (公開鍵)

の4態様がある。

【0646】これらの4態様についての処理を、認証局 (CA (PM))、パーティションマネージャ (PM)、デバイス、各エンティティ間において実行されるデータ転送処理を中心として図を用いて簡潔に説明する。

【0647】(A) 相互認証 (公開鍵)、チケット (FRT) 検証 (公開鍵)

まず、相互認証処理に公開鍵方式を適用し、チケット (FRT) 検証に公開鍵方式を適用する場合の各エンティティ間のデータ転送について図75を用いて説明する。

【0648】図に示す番号順に各エンティティ間でデータ転送が実行される。以下、各番号に従って処理を説明する。

(1) ファイル登録チケット発行手段 (FRT Issuer) の公開鍵証明書 (Cert. FRT Issuer) の発行、ファイル登録チケット発行手段 (FRT Issuer) の公開鍵証明書 (Cert. FRT Issuer) は、ファイル登録チケット発行手段 (FRT Issuer) からの発行要求により、登録局 (RA) を介した証明書発行手続きによってパーティション対応認証局 (CA (PAR)) から発行される。なお、パーティションマネージャがファイル登録チケット発行手段 (FRT Issuer) を兼ねる構成も可能であり、その場合は、ファイル登録チケット発行手段 (FRT Issuer) の公開鍵証明書としてパーティションマネージャ (PM) の公開鍵証明書を使用可能である。

【0649】(2) ファイル登録チケットユーザ (FRT User) の公開鍵証明書 (Cert. FRT User) の発行、ファイル登録チケットユーザ (FRT User: 具体的には、デバイスに対してチケットを送信するデバイスアクセス機器としてのリーダーライタ) の公開鍵証明書 (Cert. FRT User) は、ファイル登録チケットユーザ (FRT User) からの発行要求により、登録局 (RA) を介した証明書発行手続きによってパーティション対応認証局 (CA (PAR)) によって発行される。なお、パーティションマネージャがファイル登録チ

ケットユーザ (FRT User) を兼ねる構成も可能であり、その場合は、ファイル登録チケットユーザ (FRT User) の公開鍵証明書としてパーティションマネージャ (PM) の公開鍵証明書を使用可能である。

【0650】(3) ファイル登録チケット (FRT) の生成処理

ファイル登録チケット (FRT) は、パーティションマネージャの管理するファイル登録チケット発行手段 (FRT Ticket Issuer) により生成される。この場合、公開鍵方式の署名生成、検証を実行するため、ファイル登録チケット発行手段 (FRT Ticket Issuer) の秘密鍵による署名 (Signature) が生成 (図12参照) されてFRTに付加される。

【0651】(4) FRTおよびファイル登録チケット発行手段 (FRT Ticket Issuer) 公開鍵証明書 (Cert. FRT Issuer) のファイル登録チケットユーザ (FRT User) に対する供給

パーティションマネージャの管理するファイル登録チケット発行手段 (FRT Ticket Issuer) により発行されたファイル登録チケット (FRT) は、ファイル登録チケット発行手段 (FRT Ticket Issuer) 公開鍵証明書 (Cert. FRT Issuer) とともにファイル登録チケットユーザ (FRT User) すなわち、デバイスに対してチケットを送信する機器 (ex. デバイスアクセス機器としてのリーダーライタ) に対して送信される。

【0652】(5) ファイル登録チケット発行手段とデバイス間の相互認証

パーティションマネージャ (具体的にはファイル登録チケットユーザ (FRT User) であるデバイスアクセス機器としてのリーダーライタ) は、ファイル登録チケット発行手段 (FRT Ticket Issuer) の発行したファイル登録チケット (FRT) に従ったファイルを生成しようとする対象のデバイスに対し、チケットユーザ (FRT User) の公開鍵証明書 (Cert. FRT User) をデバイスに送信し、公開鍵方式の相互認証 (図50参照) を実行する。

【0653】(6) FRTおよびファイル登録チケット発行手段 (FRT Ticket Issuer) 公開鍵証明書 (Cert. FRT Issuer) のデバイスに対する供給

パーティションマネージャ (PM) とデバイス間の相互認証が成立すると、パーティションマネージャ (PM) (具体的にはファイル登録チケットユーザ (FRT User) であるデバイスアクセス機器としてのリーダーライタ) は、デバイスに対してファイル登録チケット (FRT)、およびファイル登録チケット発行手段 (FRT Ticket Issuer) 公開鍵証明書 (Cert. FRT Issuer) を送信する。

【0654】デバイスは、受信したファイル登録チケット (FRT) について、(1) チケット発行者 (Ticket Issuer) の公開鍵証明書 (CERT FRT Issuer) が改

竄されたものでない正当な公開鍵証明書 (CERT) であること、(2) チケット発行者 (Ticket Issuer) の公開鍵証明書 (CERT FRT Issuer) のオプション領域に記録されたコードと、デバイス内の PKDB (Partition Key Definition Block) (PUB) に記録されたチケット発行手段コード (FRTIC: FRT Issuer Category) の一致、(3) チケット発行手段 (Ticket Issuer) がリボークされていないこと、(4) 受信チケット (FRT) の署名 (Signature) の検証によりチケットが改竄のないことの確認を実行し、さらに、FRT チケットに格納された FRT ユーザ (チケットユーザであるデバイスアクセス機器としてのリーダライタ) とチケットユーザ (FRT User) の公開鍵証明書 (Cert. FRT User) の識別データ (DN) として記録された識別子またはカテゴリまたはシリアル (SN) 名 (DN) の一致を確認し、相互認証済みであることを確認することにより FRT ユーザ (デバイスアクセス機器としてのリーダライタ) の検証 (図 5 7、図 5 8 参照) を実行する。

【0655】(7) FDB に SPTIC および Kspt を登録

デバイスは、ファイル定義ブロック (FDB: File Definition Block) にサービス許可チケット (SPT) ユーザ (SPTIC) (ex. デバイスのファイル内のデータにアクセスを実行するデバイスアクセス機器としてのリーダライタ) と Kspt (サービス許可チケット (SPT) の MAC 検証用鍵 (Kspt)) を登録する (図 7 3 のフローにおけるステップ S827, S828)。

【0656】(8) ファイルデータ領域の確保

デバイスは、処理対象パーティションにファイル登録チケット (FRT) に記述されたサイズを持つファイル領域を確保する。

【0657】以上の処理によって、相互認証 (公開鍵)、チケット (FRT) 検証 (公開鍵) の各方式に従ったファイルの生成処理が実行される。

【0658】(B) 相互認証 (公開鍵)、チケット (FRT) 検証 (共通鍵)

次に、相互認証処理に公開鍵方式を適用し、チケット (FRT) 検証に共通鍵方式を適用する場合の各エンティティ間のデータ転送について図 7 6 を用いて説明する。

【0659】図に示す番号順に各エンティティ間でデータ転送が実行される。以下、各番号に従って処理を説明する。

【0660】(1) ファイル登録チケットユーザ (FRT User) の公開鍵証明書 (Cert. FRT User) の発行、ファイル登録チケットユーザ (FRT User: 具体的には、デバイスに対してチケットを送信するデバイスアクセス機器としてのリーダライタ) の公開鍵証明書 (Cert. FRT User) は、ファイル登録チケットユーザ (FRT User) からの発行要求により、登録局

(RA) を介した証明書発行手続きによってパーティション対応認証局 (CA (PAR)) によって発行される。なお、パーティションマネージャがファイル登録チケットユーザ (FRT User) を兼ねる構成も可能であり、その場合は、ファイル登録チケットユーザ (FRT User) の公開鍵証明書としてパーティションマネージャ (PM) の公開鍵証明書を使用可能である。

【0661】(2) ファイル登録チケット (FRT) の生成処理

ファイル登録チケット (FRT) は、パーティションマネージャの管理するファイル登録チケット発行手段 (FRT Ticket Issuer) により生成される。この場合、共通鍵方式の検証値として MAC (Message Authentication Code) (図 5 9 参照) が FRT に付加される。

【0662】(3) FRT のファイル登録チケットユーザ (FRT User) に対する供給

パーティションマネージャの管理するファイル登録チケット発行手段 (FRT Ticket Issuer) により発行されたファイル登録チケット (FRT) は、ファイル登録チケットユーザ (FRT User) すなわち、デバイスに対してチケットを送信する機器 (ex. デバイスアクセス機器としてのリーダライタ) に対して送信される。

【0663】(4) ファイル登録チケット発行手段とデバイス間の相互認証

パーティションマネージャ (具体的にはファイル登録チケットユーザ (FRT User) であるデバイスアクセス機器としてのリーダライタ) は、ファイル登録チケット発行手段 (FRT Ticket Issuer) の発行したファイル登録チケット (FRT) に従ったファイルを生成しようとする対象のデバイスに対し、チケットユーザ (FRT User) の公開鍵証明書 (Cert. FRT User) をデバイスに送信し、公開鍵方式の相互認証 (図 5 0 参照) を実行する。

【0664】(5) FRT のデバイスに対する供給

パーティションマネージャ (PM) とデバイス間の相互認証が成立すると、パーティションマネージャ (PM) (具体的にはファイル登録チケットユーザ (FRT User) であるデバイスアクセス機器としてのリーダライタ) は、デバイスに対してファイル登録チケット (FRT) を送信する。デバイスは、受信したファイル登録チケット (FRT) について MAC 検証処理を実行し、FRT 発行者 (FRT Issuer) の検証、さらに、FRT チケットに格納された FRT ユーザ (チケットユーザであるデバイスアクセス機器としてのリーダライタ) と受信したパーティションマネージャの公開鍵証明書の識別データ (DN) として記録された識別子またはカテゴリまたはシリアル (SN) 名 (DN) の一致を確認し相互認証済みであることを確認することにより FRT ユーザ (PM: デバイスアクセス機器としてのリーダライタ) の検証 (図 5 7、図 5 8 参照) を実行する。

【0665】(6) FDBにSPTICおよびKsptを登録

デバイスは、ファイル定義ブロック (FDB: File Definition Block) にサービス許可チケット (SPT) 発行者カテゴリ (SPTIC) (ex. デバイスのファイル内のデータにアクセスを実行するデバイスアクセス機器としてのリーダライタ) とKspt (サービス許可チケット (SPT) のMAC検証用鍵(Kspt)) を登録する (図73のフローにおけるステップS827, S828)。

【0666】(8) ファイルデータ領域の確保  
デバイスは、処理対象パーティションにファイル登録チケット (FRT) に記述されたサイズを持つファイル領域を確保する。

【0667】以上の処理によって、相互認証 (公開鍵)、チケット (FRT) 検証 (共通鍵) の各方式に従ったファイルの生成処理が実行される。

【0668】(C) 相互認証 (共通鍵)、チケット (FRT) 検証 (共通鍵)

次に、相互認証処理に共通鍵方式を適用し、チケット (FRT) 検証に共通鍵方式を適用する場合の各エンティティ間のデータ転送について図77を用いて説明する。

【0669】図に示す番号順に各エンティティ間でデータ転送が実行される。以下、各番号に従って処理を説明する。

【0670】(1) ファイル登録チケット (FRT) の生成処理

ファイル登録チケット (FRT) は、パーティションマネージャの管理するファイル登録チケット発行手段 (FRT Ticket Issuer) により生成される。この場合、共通鍵方式の検証値としてMAC (Message Authentication Code) (図59参照) がFRTに付加される。

【0671】(2) FRTのファイル登録チケットユーザ (FRT User) に対する供給

パーティションマネージャの管理するファイル登録チケット発行手段 (FRT Ticket Issuer) により発行されたファイル登録チケット (FRT) は、ファイル登録チケットユーザ (FRT User) すなわち、デバイスに対してチケットを送信する機器 (ex. デバイスアクセス機器としてのリーダライタ) に対して送信される。

【0672】(3) ファイル登録チケット発行手段とデバイス間の相互認証

パーティションマネージャ (具体的にはファイル登録チケットユーザ (FRT User) であるデバイスアクセス機器としてのリーダライタ) は、ファイル登録チケット発行手段 (FRT Ticket Issuer) の発行したファイル登録チケット (FRT) に従ったファイルを生成しようとする対象のデバイスとの間で、共通鍵方式の相互認証 (図53、図54参照) を実行する。

【0673】(4) FRTのデバイスに対する供給

パーティションマネージャ (PM) とデバイス間の相互認証が成立すると、パーティションマネージャ (PM)

(具体的にはファイル登録チケットユーザ (FRT User) であるデバイスアクセス機器としてのリーダライタ) は、デバイスに対してファイル登録チケット (FRT) を送信する。デバイスは、受信したファイル登録チケット (FRT) についてMAC検証処理を実行し、FRT発行者 (FRT Issuer) の検証、さらに、FRTチケットに格納されたFRTユーザ (チケットユーザであるデバイスアクセス機器としてのリーダライタ) と受信したパーティションマネージャの識別子の一致を確認し相互認証済みであることを確認することによりFRTユーザ (PM: デバイスアクセス機器としてのリーダライタ) の検証 (図57、図58参照) を実行する。

【0674】(6) FDBにSPTICおよびKsptを登録

デバイスは、ファイル定義ブロック (FDB: File Definition Block) にサービス許可チケット (SPT) 発行者カテゴリ (SPTIC) (ex. デバイスのファイル内のデータにアクセスを実行するデバイスアクセス機器としてのリーダライタ) とKspt (サービス許可チケット (SPT) のMAC検証用鍵(Kspt)) を登録する (図73のフローにおけるステップS827, S828)。

【0675】(8) ファイルデータ領域の確保  
デバイスは、処理対象パーティションにファイル登録チケット (FRT) に記述されたサイズを持つファイル領域を確保する。

【0676】以上の処理によって、相互認証 (共通鍵)、チケット (FRT) 検証 (共通鍵) の各方式に従ったファイルの生成処理が実行される。

【0677】(D) 相互認証 (共通鍵)、チケット (FRT) 検証 (公開鍵)

次に、相互認証処理に共通鍵方式を適用し、チケット (FRT) 検証に公開鍵方式を適用する場合の各エンティティ間のデータ転送について図78を用いて説明する。

【0678】図に示す番号順に各エンティティ間でデータ転送が実行される。以下、各番号に従って処理を説明する。

(1) ファイル登録チケット発行手段 (FRT Issuer) の公開鍵証明書 (Cert. FRT Issuer) の発行、ファイル登録チケット発行手段 (FRT Issuer) の公開鍵証明書 (Cert. FRT Issuer) は、ファイル登録チケット発行手段 (FRT Issuer) からの発行要求により、登録局 (RA) を介した証明書発行手続きによってパーティション対応認証局 (CA (PAR)) から発行される。なお、パーティションマネージャがファイル登録チケット発行手段 (FRT Issuer) を兼ねる構成も可能であり、その場合は、ファイル登録チケット発行手段 (FRT Issuer) の公開鍵証明書と

してパーティションマネージャ (PM) の公開鍵証明書を使用可能である。

【0679】(2) ファイル登録チケット (FRT) の生成処理

ファイル登録チケット (FRT) は、パーティションマネージャの管理するファイル登録チケット発行手段 (FRT Ticket Issuer) により生成される。この場合、公開鍵方式の署名生成、検証を実行するため、ファイル登録チケット発行手段 (FRT Ticket Issuer) の秘密鍵による署名 (Signature) が生成 (図12参照) されて FRT に付加される。

【0680】(3) FRTおよびファイル登録チケット発行手段 (FRT Ticket Issuer) 公開鍵証明書 (Cert. FRT Issuer) のファイル登録チケットユーザ (FRT User) に対する供給

パーティションマネージャの管理するファイル登録チケット発行手段 (FRT Ticket Issuer) により発行されたファイル登録チケット (FRT) は、ファイル登録チケット発行手段 (FRT Ticket Issuer) 公開鍵証明書 (Cert. FRT Issuer) とともにファイル登録チケットユーザ (FRT User) すなわち、デバイスに対してチケットを送信する機器 (ex. デバイスアクセス機器としてのリーダライタ) に対して送信される。

【0681】(4) ファイル登録チケット発行手段とデバイス間の相互認証

パーティションマネージャ (具体的にはファイル登録チケットユーザ (FRT User) であるデバイスアクセス機器としてのリーダライタ) は、ファイル登録チケット発行手段 (FRT Ticket Issuer) の発行したファイル登録チケット (FRT) に従ったファイルを生成しようとする対象のデバイスとの間で、共通鍵方式の相互認証 (図53、図54参照) を実行する。

【0682】(5) FRTおよびファイル登録チケット発行手段 (FRT Ticket Issuer) 公開鍵証明書 (Cert. FRT Issuer) のデバイスに対する供給  
パーティションマネージャ (PM) とデバイス間の相互認証が成立すると、パーティションマネージャ (PM) (具体的にはファイル登録チケットユーザ (FRT User) であるデバイスアクセス機器としてのリーダライタ) は、デバイスに対してファイル登録チケット (FRT)、およびファイル登録チケット発行手段 (FRT Ticket Issuer) 公開鍵証明書 (Cert. FRT Issuer) を送信する。

【0683】デバイスは、受信したファイル登録チケット (FRT) について、(1) チケット発行者 (Ticket Issuer) の公開鍵証明書 (CERT FRT Issuer) が改竄されたものでない正当な公開鍵証明書 (CERT) であること、(2) チケット発行者 (Ticket Issuer) の公開鍵証明書 (CERT FRT Issuer) のオプション領域に記録されたコードと、デバイス内のPKDB (Part

ition Key DefinitionBlock) (PUB) に記録されたチケット発行手段コード (FRTIC: FRT Issuer Category) の一致、(3) チケット発行手段 (Ticket Issuer) がリボークされていないこと、(4) 受信チケット (FRT) の署名 (Signature) の検証によりチケットが改竄のないことの確認を実行し、さらに、FRTチケットに格納されたFRTユーザ (チケットユーザであるデバイスアクセス機器としてのリーダライタ) とチケットユーザ (FRT User) の識別子の一致を確認し、相互認証済みであることを確認することによりFRTユーザ (デバイスアクセス機器としてのリーダライタ) の検証 (図57、図58参照) を実行する。

【0684】(6) FDBにSPTICおよびKsptを登録

デバイスは、ファイル定義ブロック (FDB: File Definition Block) にサービス許可チケット (SPT) 発行者カテゴリ (SPTIC) (ex. デバイスのファイル内のデータにアクセスを実行するデバイスアクセス機器としてのリーダライタ) とKspt (サービス許可チケット (SPT) のMAC検証用鍵 (Kspt)) を登録する (図73のフローにおけるステップS827、S828)。

【0685】(7) ファイルデータ領域の確保  
デバイスは、処理対象パーティションにファイル登録チケット (FRT) に記述されたサイズを持つファイル領域を確保する。

【0686】以上の処理によって、相互認証 (共通鍵)、チケット (FRT) 検証 (公開鍵) の各方式に従ったファイルの生成処理が実行される。

【0687】[B4. 6. サービス許可チケット (SPT) を利用したサービス (ファイルアクセス) 処理] 次に、サービス許可チケット (SRT) (図28、図31参照) を利用したファイルアクセス処理について説明する。図79以下のフロー他の図面を参照して説明する。なお、ファイルアクセス処理には、デバイスとファイルアクセス装置間における相互認証処理 (デバイス認証またはパーティション認証)、サービス許可チケット (SPT: Service Permission Ticket) の正当性検証処理が含まれる。

【0688】図79のフローにおいて、左側がファイルアクセス装置、右側がデバイス (図5参照) の処理を示す。なお、ファイルアクセス装置は、パーティションマネージャの管理装置であり、デバイスに対するデータ読み取り書き込み処理可能な装置 (ex. デバイスアクセス機器としてのリーダライタ、PC) であり、図10のデバイスアクセス機器としてのリーダライタに相当する。まず、図79を用いて、ファイルアクセス装置によるファイルアクセス処理の概要を説明し、その後、当処理に含まれる各処理の詳細を図80以下のフローを用いて順次説明する。

【0689】まず、図79のステップS851とS86

0において、ファイルアクセス装置とデバイス間にでの相互認証処理が実行される。データ送受信を実行する2つの手段間では、相互に相手が正しいデータ通信者であるか否かを確認して、その後に必要なデータ転送を行なうことが行われる。相手が正しいデータ通信者であるか否かの確認処理が相互認証処理である。相互認証処理時にセッション鍵の生成を実行して、生成したセッション鍵を共有鍵として暗号化処理を実行してデータ送信を行なう。

【0690】相互認証処理については、先のパーティション生成、削除処理の欄で説明したと同様の処理であり、パーティション認証が実行される。それぞれについて共通鍵方式認証、あるいは公開鍵方式認証処理のいずれかが適用される。この相互認証処理は、前述の図48～図56を用いて説明したと同様の処理であるので説明を省略する。

【0691】なお、相互認証処理として実行すべき処理は、適用するサービス許可チケット（SPT）（図28、図31参照）の

\* Authentication Flag : チケット（Ticket）の利用処理においてデバイス（Device）との相互認証が必要か否かを示すフラグ

\* Authentication Type : デバイス（Device）の相互認証のタイプ（公開鍵認証、または、共通鍵認証、または、いずれでも可（Any））によって決定される。

【0692】認証処理に失敗した場合（S852、S861でNo）は、相互が正当な機器、デバイスであることの確認がとれないことを示し、以下の処理は実行されずエラーとして処理は終了する。

【0693】デバイスは、複数のサービス許可チケット（SPT）に基づく複数の異なるパーティション内のファイルアクセスを許容する処理も可能である。例えば、デバイス認証の成立を条件として、複数のサービス許可チケット（SPT）に基づく複数の異なるパーティション内のファイルアクセスを許容することが可能である。各パーティション毎のファイルアクセスルールは、アクセス制御データとして構成されるサービス許可チケット（SPT）に記述され、デバイスは、アクセス機器から複数のサービス許可チケット（SPT）を受領し、各チケットがデバイス認証を要求している場合は、記述に従ってデバイス認証が成立したことを条件として各パーティション内のファイルアクセスを許容する。

【0694】また、デバイスは、複数のサービス許可チケット（SPT）の各々が異なる認証条件を定めている場合は、各サービス許可チケット（SPT）に設定されたパーティション認証の認証成立を条件として、複数のサービス許可チケット（SPT）の指定ファイルに対するアクセスを許容する。

【0695】次にステップS853において、ファイル

アクセス装置は、デバイスに対してサービス許可チケット（SPT : Service Permission Ticket）を送信する。サービス許可チケット（SPT）は、パーティションマネージャの管理下のサービス許可チケット（SPT）発行手段（SPT Issuer）により発行されるチケットである。サービス許可チケット（SPT）は、デバイスに対するアクセス制御チケットであり、先に説明した図28、図31のデータフォーマット構成を持つチケットである。

【0696】なお、サービス許可チケット（SPT）を、チケットユーザに対して送信する際には、公開鍵方式の場合、サービス許可チケット（SPT）発行手段（SPT Issuer）の公開鍵証明書（CERT\_SPTI）も一緒に送信する。SPT発行手段の公開鍵証明書（CERT\_SPTI）の属性（Attribute）は、デバイス内のFDB（File Definition Block）に記録されたチケット発行手段コード（SPTIC）と一致する。

【0697】ファイル登録チケット（SPT）を受信（S862）したデバイスは、受信したチケット（SRT）の正当性と利用者チェック処理を実行（S863）する。チケットの正当性の検証処理は、共通鍵方式によるMAC検証、あるいは公開鍵方式による署名検証処理のいずれかを適用して実行される。利用者チェックは、チケットを送信してきた機器（チケット利用者）の正当性をチェックする処理であり、相互認証が成立済みであり、認証相手の識別データと、チケットに記録されているチケットユーザ識別子（図28、図31参照）との一致等を検証する処理として実行される。これらの処理は、先のパーティション登録チケット（PRT）の適用処理についての説明中、図57～図59を用いて説明したと同様の処理であるので説明を省略する。

【0698】デバイスにおいて、受信チケット（SPT）の正当性と利用者チェック処理の結果、チケットおよび利用者の正当なことが確認できなかった場合（S864でNo）は、サービス許可チケット（SPT）受理エラーをファイルアクセス装置に通知（S868）する。チケットおよび利用者の正当なことが確認できた場合（S864でYes）は、受信したサービス許可チケット（SPT）に記述されたルールに従いデバイス内のメモリ部に格納されたファイルオープン処理が実行される。この処理の詳細については、別フローを用いて後段で詳述する。

【0699】サービス許可チケット（SPT）の記述に従って、ファイルのオープン処理に成功（S866でYes）すると、SPT受理成功をファイルアクセス装置に通知（S867）する。一方、ファイルのオープン処理に失敗（S866でNo）した場合は、SPT受理エラーをファイルアクセス装置に通知（S868）する。

【0700】ファイルアクセス装置は、SPT受理結果を受信（S854）し、SPT処理結果を判定し、SP

T受理結果がエラーである場合（S855でNo）は、エラーとして処理を終了し、SPT受理結果が成功（S855でYes）である場合は、すべてのSPT送信が終了したか否かを判定（S856）し、未送信SPTがある場合は、ステップS853以下を繰り返し実行する。

【0701】すべてのSPT送信が終了した場合は、ステップS857、S869においてサービス許可チケット（SPT）に従ったファイルアクセスを実行し、ファイルアクセス処理の終了の後、セッションクリアコマンドの送受信（S858、S870）を実行し、デバイス側に生成した認証テーブルを破棄（S871）し、処理を終了する。ファイルアクセス処理の詳細については、別フローを用いて後段で詳述する。なお、認証テーブルは、ステップS851、S860の相互認証処理において生成されるテーブルであり、前述したパーティション登録チケット（PRT）の適用処理の項目において説明した構成、すなわち、図51の構成と同様のものである。

【0702】このようにサービス許可チケット（SPT）を利用して、デバイス内に設定されたパーティション内のファイルに対するアクセス処理が実行される。以下、当処理に含まれるファイルオープン処理（S865）、各種ファイルアクセス処理（S857、S869）について説明する。

【0703】（ファイルオープン処理）図80のフローに従って、ファイルオープン処理（図79、S865）について説明する。ファイルオープン処理は、デバイスが受信したサービス許可チケット（SPT）に従って実行する処理である。

【0704】ステップS881において、デバイスは、受信したサービス許可チケット（SPT）に指定されたファイルがデバイス内に生成されて存在しているか否かを判定する。サービス許可チケット（SPT）には処理対象ファイルのファイルIDが記録（図28、図31参照）されており、同一のIDを持つファイルの有無を例えばファイル定義ブロック（図24）を参照して判定する。チケットに記述されたIDと同一IDのファイルが存在しない場合は、処理が不可能であるのでエラー終了する。

【0705】チケットに記述されたIDと同一IDのファイルが存在する場合は、ステップS882において、ファイルオープンテーブルにサービス許可チケット（SPT）に記述されたチケット発行手段（Ticket issuer = PMC: Partition manager Code）と、サービス許可チケット（SPT）に記述されたファイルIDとを対応付けたエントリを書き込む。

【0706】さらに、ステップS883において、ファイルオープンテーブルに生成したエントリに対応付けてサービス許可チケット（SPT）に記述されたファイル

アクセスモード [File Access Mode : アクセスを許諾するファイル (File) へのアクセスモード (Access Mode)] を書き込み、ステップS884において、サービス許可チケット（SPT）に記述されたアクセス許可グループファイル [Group of Target File : アクセスを許すファイル (File) のグループ (Group)] を書き込み、ステップS885において、サービス許可チケット（SPT）に記述されたアクセス許可ファイル識別子 [Target File ID : アクセスを許すファイル (File) の識別子 (ID)] の書き込み、さらにステップS886において、サービス許可チケット（SPT）に記述されたターゲットファイル (Target File) に対する処理態様データ [Read/Write Permission : アクセスを許すファイル (File) (ターゲットファイル (Target File)) に対する処理態様 (読み出し (Read), 書き込み (Write) の許可)] の書き込み処理を行なう。なお、ターゲットファイルに対する処理としては、読み出し (Read), 書き込み (Write) に限らず、様々な処理を設定可能である。

【0707】ファイルオープンテーブルの構成例を図81、図82に示す。ファイルオープンテーブルは、デバイスにおいてアクセス処理状態にあるファイルおよびアクセスモード他の情報を記録したテーブルであり、デバイスが受信したサービス許可チケット（SPT）の記述情報を記録してデバイスの記憶手段に格納する。

【0708】チケットが唯一のファイルに対してのみアクセスを許可する形式のサービス許可チケット（図28参照）である場合は、ファイルオープンテーブルは、  
\* Ticket Issuer : チケット発行手段 (Ticket Issuer) の識別子

\* File ID : パーティション内のアクセスファイル (File) の識別子 (ID)

\* File Access Mode : アクセスを許諾するファイル (File) へのアクセスモード (Access Mode)

の情報を格納する。この場合のファイルオープンテーブルの構成例を図81に示す。

【0709】図81に示すように、ファイルオープンテーブルにはグループ情報であるTicket Issuer : チケット発行手段 (Ticket Issuer) の識別子として、パーティションマネージャコード (PMC) が記述され、パーティションが判別され、ファイルIDによりファイルが識別され、ファイルアクセスモードにより実行可能なアクセス態様 (ex. 読み取り (READ), 書き込み (Write), 暗号化復号化 (Enc, Dec)) が判定可能となる。

【0710】また、サービス許可チケット（SPT）がパーティションに設定されたファイル中の複数ファイルに対してアクセスを許可する形式のサービス許可チケット（図31参照）の場合は、上記情報に加えて

\* Group of Target File : アクセスを許すファイル (Fi

le) のグループ (Group)

\* Target File ID : アクセスを許すファイル (File) の識別子 (ID)

\* Read/Write Permission : アクセスを許すファイル (File) (ターゲットファイル (Target File)) に対する処理態様 (読み出し (Read), 書き込み (Write)) の許可

の各情報がテーブルに書き込まれる。この場合のファイルオープンテーブルの構成例を図 8 2 に示す。

【0711】図 8 2 に示すように、複数ファイルに対してアクセスを許可する形式のサービス許可チケットに対応して設定されるファイルオープンテーブルには、図 8 1 に示すデータの他に、アクセスを許すターゲットファイル (File) のグループとしてのパーティション識別データとしてのパーティションマネージャコード (PMC) と、アクセスを許すターゲットファイル (File) の識別子 (ID) としてのファイル ID と、ターゲットファイル (Target File) に対する処理態様を示す [Read/Write Permission] データが格納され、複数ファイルに対する実行可能な処理が判定可能となる。

【0712】複数のファイルに対してアクセスを実行する処理とは、例えばファイル A に格納された鍵を用いて、ファイル B に格納されたデータを暗号化する処理を実行する場合などである。このためには、ファイル B はファイル A の読み出し要求に対して許可を与える必要がある。この場合、ファイル B のことをソースファイル、許可を与える相手のファイルのことをターゲットファイルと呼ぶ。

【0713】このように、デバイスは、アクセス機器とのセッション中に受領したサービス許可チケット (SPT) に基づいて、チケット発行手段 (Ticket Issuer (PM C)) としてのパーティションマネージャコード (PMC)、ファイルオープン処理を実行したファイルの識別データとしてのファイル識別子と、サービス許可チケット (SPT) に記述されたアクセスモードを対応付けたファイルオープンテーブルを生成し、該ファイルオープンテーブルを参照して前記アクセス機器からの受領コマンドの実行の可否の判定が可能となる。

【0714】(ファイルアクセス処理) 次に、図 7 9 のステップ S 8 5 7、S 8 6 9 において実行されるファイルアクセス処理の詳細について説明する。

【0715】まず、図 8 1 に示すファイルオープンテーブルが生成された場合のアクセス処理について、図 8 3 を用いて説明する。図の左側に 2 つのファイルアクセス装置 (R/W : デバイスアクセス機器としてのリーダライタ) 7 5 0、7 6 0 を示し、右側にファイルの生成されたデバイス 1 0 0 のパーティション部分を示す。

【0716】ファイルアクセス装置 (R/W : リーダライタ) 7 5 0 は、デバイスとの相互認証の後、ファイル ID : [0 x 0 0 0 2]

ファイルアクセスモード : [読み取り : Read]

のアクセス許可チケットをデバイス 1 0 0 に送信し、チケットの正当性検証、チケット発行者、利用者検証に成功したとする。

【0717】このとき、デバイスには図 8 1 に示すファイルオープンテーブルの第 2 行のエントリが生成される。このエントリは、パーティションマネージャコード (PMC 1) で識別されるパーティション内のファイル ID [0 x 0 0 0 2] に対してアクセスモード [読み取り : Read] の処理が実行可能であることを示している。

【0718】このとき、ファイルアクセス装置 (R/W : リーダライタ) 7 5 0 は、コマンドを生成してデバイスに対して送信する。例えばファイル ID [0 x 0 0 0 2] のデータ読み取りコマンド : Read Command (0 x 0 0 0 2) をデバイスが受信すると、デバイスはファイルオープンテーブルのエントリを確認し、ファイル ID [0 x 0 0 0 2] に対してアクセスモード [読み取り : Read] の処理が実行可能であることを確認し、読み取り処理を実行する。

【0719】また、ファイルアクセス装置 (R/W : リーダライタ) 7 5 0 が、例えばファイル ID [0 x 0 0 0 2] のデータ書き込みコマンド : Write Command (0 x 0 0 0 2)、あるいはファイル ID [0 x 0 0 0 1] のデータの暗号化処理コマンド : Encryption Command (0 x 0 0 0 1) をデバイスに送信した場合は、コマンドを受信したデバイスはファイルオープンテーブルのエントリを確認し、ファイル ID [0 x 0 0 0 2] に対する [書き込み : Write] の処理、および、ファイル ID [0 x 0 0 0 1] の [暗号化処理] が、ファイルアクセス装置 (R/W : リーダライタ) 7 5 0 から受領したサービス許可チケット (SPT) によって許可されていないことを確認し、処理を停止する。

【0720】また、ファイルアクセス装置 (R/W : リーダライタ) 7 6 0 は、デバイスとの相互認証の後、ファイル ID : [0 x 0 0 0 1]

ファイルアクセスモード : [暗号化復号化処理 : Enc & Dec]

のアクセス許可チケットをデバイス 1 0 0 に送信し、チケットの正当性検証、チケット発行者、利用者検証に成功したとする。

【0721】このとき、デバイスには図 8 1 に示すファイルオープンテーブルの第 1 行のエントリが生成される。このエントリは、パーティションマネージャコード (PMC 1) で識別されるパーティション内のファイル ID [0 x 0 0 0 1] に対してアクセスモード [暗号化復号化処理 : Enc & Dec] の処理が実行可能であることを示している。

【0722】このとき、ファイルアクセス装置 (R/W :

W:リーダライタ) 760は、コマンドを生成してデバイスに対して送信する。例えばファイルID [0x0001] の暗号化コマンド [Encryption Command (0x0001)] をデバイスが受信すると、デバイスはファイルオープンテーブルのエントリを確認し、ファイルID: 0x0001に対してアクセスモード [暗号化復号化処理: Enc&Dec] の処理が実行可能であることを確認し、暗号化処理を実行する。

【0723】また、ファイルアクセス装置 (R/W:リーダライタ) 760が、例えばファイルID [0x0002] のデータ読み取りコマンド: Read Command (0x0002) をデバイスに送信した場合は、コマンドを受信したデバイスはファイルオープンテーブルのエントリを確認し、ファイルID [0x0002] に対する [読み取り: Read] の処理が、ファイルアクセス装置 (R/W:リーダライタ) 760から受領したサービス許可チケット (SPT) によって許可されていないことを確認し、処理を停止する。

【0724】このように、サービス許可チケットユーザであるデバイスアクセス機器としてのリーダライタからデバイスが受信したサービス許可チケット (SPT) に基づいて、デバイスは、前述の図80の処理フローに従ってファイルオープンテーブルを生成し、生成したファイルオープンテーブルに基づいてファイルアクセス装置であるリーダライタからの各コマンドの実行可否を決定し、決定に従って処理を実行する。

【0725】次に、2つのファイルに対する処理を実行する場合のアクセス処理について、図84を用いて説明する。図の左側に2つのファイルアクセス装置 (R/W:リーダライタ) 770、780を示し、右側にファイルの生成されたデバイス100のパーティション部分を示す。

【0726】まず、ターゲットファイルを指定したサービス許可チケット (SPT) (図31参照) を使用した処理の実行例を説明する。

【0727】ファイルアクセス装置 (R/W:リーダライタ) 770は、デバイスとの相互認証の後、SPTフォーマット1

ファイルID: [0x0001]

ファイルアクセスモード: [暗号化復号化処理: Enc&Dec]

および、SPTフォーマット2

ファイルID: [0x0002]

ターゲットファイル・グループ: [PMC1]

ターゲットファイルID: [0x0001]

読書き許可: [読み取り: Read]

の2つのアクセス許可チケットをデバイス100に送信し、チケットの正当性検証、チケット発行者、利用者検証に成功したとする。

【0728】このとき、デバイスには図82に示すフ

イルオープンテーブルのエントリが生成される。このエントリは、パーティションマネージャコード (PMC1) で識別されるパーティション内のファイルID [0x0001] は、鍵のファイルであり、暗号化、復号化が可能になるようにオープンされる。ファイルID [0x0002] は、データのファイルであり、外部からの読み出しはファイルアクセスモード (File Access Mode) の欄が空白のため不可能であり、ファイルID [0x0001] に対して [読み取り: Read] を実行可能とする目的のためにオープンされて、ファイルオープンテーブルのエントリとして設定されていることを示している。

【0729】このとき、ファイルアクセス装置 (R/W:リーダライタ) 770は、コマンドを生成してデバイスに対して送信する。例えばファイルID [0x0002] の読み取り、ファイルID [0x0001] による内部暗号化コマンド: Internal Encryption Command (0x0001, 0x0002) を送信し、デバイスがコマンドを受信すると、デバイスはファイルオープンテーブルのエントリを確認し、ファイルID [0x0002] に対して、ターゲットファイルグループ [PMC1]、ターゲットファイル [0x0001] の [暗号化処理] に対して [読み取り: Read] を実行可能であることを判定し、ファイルID [0x0002] のデータを読み取り、ファイルID [0x0001] の鍵 (key) による暗号化を実行して暗号化データをアクセス装置に送信する。

【0730】このターゲットファイルを指定したサービス許可チケット (SPT) (図31参照) を使用した処理によればあるファイルから読み出したデータを他のファイルに格納された暗号化鍵を用いて暗号化したデータを取得する処理が可能となり、復号データが外部に漏れるおそれがない。

【0731】次に、ターゲットファイルを指定したサービス許可チケット (SPT) (図31参照) ではなく、唯一のファイルに対する処理を指定したサービス許可チケット (SPT) (図28参照) を複数用いた場合の処理について説明する。

【0732】ファイルアクセス装置 (R/W:リーダライタ) 780は、デバイスとの相互認証の後、SPTフォーマット1として、

ファイルID: [0x0002]

ファイルアクセスモード: [読み取り: Read]

またSPTフォーマット2として、

ファイルID: [0x0001]

ファイルアクセスモード: [暗号化復号化処理: Enc&Dec]

の2つのアクセスチケットをデバイス100に送信し、チケットの正当性検証、チケット発行者、利用者検証に成功したとする。



【0733】このとき、デバイスには図81に示すファイルオープンテーブルの第1行および第2行の各エントリが生成される。このエントリは、パーティションマネージャコード（PMC1）で識別されるパーティション内のファイルID [0x0001] に対してアクセスモード [暗号化復号化処理：Enc&Dec] の処理が実行可能であり、パーティションマネージャコード（PMC1）で識別されるパーティション内のファイルID：0x0002に対してアクセスモード [読み取り：Read] の処理が実行可能であることを示している。

【0734】このとき、ファイルアクセス装置（R/W：リーダライタ）780は、コマンドを生成してデバイスに対して送信する。まず、ファイルID：[0x0002] のデータ読み取りコマンド：Read Command (0x0002) をデバイスが受信すると、デバイスはファイルオープンテーブルのエントリを確認し、ファイルID：0x0002に対してアクセスモード [読み取り：Read] の処理が実行可能であることを確認し、読み取り処理を実行し、ファイルアクセス装置に読み取りデータを送信する。

【0735】次に、ファイルアクセス装置（R/W：リーダライタ）780は、さらにコマンドを生成してデバイスに対して送信する。ファイルID [0x0001] によるデータ（Data）の暗号化コマンド [Encryption Command (0x0001, Data)] をデバイスが受信すると、デバイスはファイルオープンテーブルのエントリを確認し、ファイルID [0x0001] に対してアクセスモード [暗号化復号化処理：Enc&Dec] の処理が実行可能であることを確認し、暗号化処理を実行し暗号化データ [Encryption Data] をファイルアクセス装置（R/W：リーダライタ）780に送信する。

【0736】このように、ターゲットファイルを指定したサービス許可チケット（SPT）（図31参照）ではなく、唯一のファイルに対する処理を指定したサービス許可チケット（SPT）（図28参照）を複数用いた場合は、暗号化対象データの読み出し処理が実行され、ファイルアクセス装置780とデバイス間におけるデータ転送処理の回数が増加することになる。また、データが暗号化されずにデバイスの外に読み出されてしまう。

【0737】一方、サービス許可チケット（SPT）（図31参照）に、アクセス対象とした複数のデータファイルを識別する複数のファイル識別子を含ませ、該複数のファイル識別子中、一方はターゲットファイル識別子として設定するとともにターゲットファイルに対する読み取りまたは書き込み許可データを格納し、他方のデータファイルのアクセスモードとして該データファイルに格納した暗号鍵を用いた暗号化処理を設定した構成とすれば、メモリ搭載デバイスは、アクセス機器からサービス許可チケット（SPT）を受領して、指定アクセス

モードに従った処理として、ターゲットファイルの読み取りおよび暗号鍵による暗号化処理を実行することになり、メモリ搭載デバイス内における内部暗号化処理を実行することが可能となる。また、データが暗号化されずにデバイスの外に流出することを防止することができる。

【0738】サービス許可チケット（SPT）を発行するチケット発行手段は、メモリ搭載デバイスのメモリ領域を管理するエンティティであるパーティションマネージャの管理下にあるチケット発行手段であり、各アクセス機器に応じて各種のアクセスモードを設定したサービス許可チケット（SPT）を個別に発行することにより、各アクセス機器に応じて異なる態様のアクセスを実行可能とした構成が実現される。

【0739】（セッション鍵の使用態様）なお、ファイルアクセス装置とデバイス間において送受信されるデータは、デバイスのユーザ情報、あるいは金額情報など外部に対する漏洩を防止すべきデータであることが多い。従ってファイルアクセス装置とデバイス間において送受信されるデータは、暗号化処理を実行し、また改竄チェック値としてのMAC（Message Authentication Code）を付加したデータとするのが望ましい。

【0740】データの暗号化には、ファイルアクセス装置とデバイス間において実行される相互認証処理において生成されるセッション鍵を用いることが可能である。前述したように相互認証には、デバイスに対するデバイス認証、各パーティションに対する認証としてのパーティション認証がある。パーティションに生成済みのファイルに対するアクセスを実行する場合、データ転送の際に適用する暗号化鍵としていずれを適用するかはいくつかの選択肢がある。

【0741】例えば図85に示すように、デバイス100とアクセス装置800との間において、デバイス認証によって生成されたセッション鍵Kses1、パーティションマネージャコード（PMC1）に対応するパーティションとのパーティション認証によって生成されたセッション鍵Kses2、パーティションマネージャコード（PMC2）に対応するパーティションとのパーティション認証によって生成されたセッション鍵Kses3がある場合がある。

【0742】これらは、相互認証の際に生成される認証テーブル（図51、図52参照）にセッションクリアまで格納される。

【0743】デバイスとデバイスとの通信を実行するデバイスアクセス機器としてのリーダライタ（PC他の通信装置）は、これら複数のセッション鍵のいずれを適用して暗号化通信を実行するかを、ルールとして予め取り決めておき、決定されたルールに従ってセッション鍵を適用する構成が可能である。

【0744】複数の異なるパーティション内のファイル

アクセスを、複数の異なるパーティション各々に対応して設定された認証条件であるパーティション認証またはデバイス認証のすべての認証成立を条件として許容する場合、複数の認証処理の結果、取得された複数のセッション鍵に基づいて唯一の統合セッション鍵を生成し、該統合セッション鍵に基づいてアクセス機器との通信データの暗号化処理を実行する。

【0745】統合セッション鍵生成手法としての1つの方法は、デバイスとデバイスとの通信を実行するデバイスアクセス機器としてのリーダライタ（PC他の通信装置）間における相互認証処理によって複数のセッション鍵 $Kses1 \sim KsesN$ が生成された場合、これら複数のセッション鍵 $Kses1 \sim KsesN$ の排他論理処理（ex. 8バイト処理）を実行し演算結果を通信データの暗号化用セッション鍵とする方法である。すなわち、

$Kses = Kses1 \text{ XOR } Kses2 \text{ XOR } Kses3 \dots$

XOR：排他論理処理（ex. 8バイト処理）

によって算出された $Kses$ をセッション鍵として用いる。

【0746】デバイスとデバイスとの通信を実行するデバイスアクセス機器としてのリーダライタ（PC他の通信装置）間では、双方の認証テーブルに格納されたセッション鍵を排他論理処理と演算しその出力値をセッション鍵として使用するというルールを定め、該ルールに基づいてセッション鍵を算出して通信データの暗号化に用いる構成とする。また、同様に、相互認証時に同時に共有した別のセッション鍵、例えば公開鍵認証時において生成したセッション鍵、あるいはセッション鍵生成データ、例えばY座標の下位64ビットを用いてセッション中の通信データにMAC値を付加することができる。このMAC値を通信データ（暗号化データである場合もある）とともに送信し、受信側でMAC検証処理を行なうことで、通信路上のデータ改竄を防止することが可能となる。MAC生成、検証処理については先に説明した図59を参照されたい。

【0747】あるいは、デバイスとデバイスとの通信を実行するデバイスアクセス機器としてのリーダライタ

（PC他の通信装置）間における相互認証処理によって取得された複数のセッション鍵 $Kses1 \sim KsesN$ の中で、いずれかの1つの鍵（ex. 最新のセッション鍵）を選択してその後の通信処理におけるデータ暗号化鍵として適用するというルールを設定し、該ルールに従って、セッション鍵を選択して通信データの暗号化に用いる構成としてもよい。

【0748】なお、上述した複数のセッション鍵の演算による算出、あるいは選択処理は、ファイルアクセス装置とデバイス間の暗号化通信のみならず、すべてのチケット（PRT, FRT, SPT, DUT）ユーザ（デバイスアクセス機器としてのリーダライタなどのデバイス

とのデータ通信を実行する機器）とデバイス間の暗号化通信処理において、相互認証により複数のセッション鍵が生成された場合に適用できる。各チケットユーザとデバイス相互において、どのようなルールに従って、適用するセッション鍵を複数のセッション鍵から算出するかまたは選択するかについては予めルールとして取り決め、相互にルールを確認した後実行するか、あるいは各チケットにルールを記録しておくなどの措置が採用可能である。

【0749】次に、図86、図87を用いてファイルアクセス装置によるデバイスに対するアクセス処理（図79の処理フローにおけるステップS857、869）手順の代表的例について説明する。

【0750】図86を用いて1つのファイルのみに対してアクセスを実行する場合の処理（Normal）について説明し、図87を用いて複数ファイルに対してアクセスを行なう場合の処理（Combination）について説明する。

【0751】まず、図86の1つのファイルのみに対してアクセスを実行する場合の処理（Normal）について説明する。図86のフローにおいて、左側がファイルアクセス装置、右側がデバイス（図5参照）の処理を示す。なお、ファイルアクセス処理において、ファイルアクセス装置とデバイス間におけるデータ転送の際には、相互認証処理において取得したセッション鍵 $Kses$ 、あるいは複数のセッション鍵から演算または選択されたセッション鍵を用いて暗号化され、また改竄チェック用のMACの生成、検証処理が実行される。

【0752】ファイルアクセス装置は、ステップS891において、アクセスコマンドをデバイスに送信する。このコマンドは、アクセス対象のファイルID、アクセスモードを指定したコマンドであり、例えば先に図83を用いて説明したようなファイルID[0x0002]のデータ読み取りコマンド：Read Command（0x0002）、あるいは、ファイルID[0x0001]の暗号化コマンド[Encryption Command（0x0001）]などである。

【0753】デバイスは、ファイルアクセス装置からのコマンドを受信（S901）すると、コマンドに含まれるファイルID、アクセスモードがファイルオープンテーブルに許可されたエントリとして記録されているかを判定（S902）する。ファイルオープンテーブルにコマンドに対応するファイルID、アクセスモードのエントリが存在しない場合は、コマンドに従った処理を実行せず、アクセスエラーをファイルアクセス装置に送信（S908）する。

【0754】ファイルオープンテーブルにコマンドに対応するファイルID、アクセスモードのエントリが存在した場合は、ステップS903において、デバイスのメモリ内の対応パーティションのファイル定義ブロック

(FDB) (図24参照)に記録されたファイルアクセス認証方式 (Acceptable Authentication Type :特定のファイル (File) アクセスを行う際に、指定する認証方式) を参照し、アクセス対象のファイルに対するアクセスコマンドの実行に必要な認証レベル (公開鍵認証を必要とするか) を確認する。

【0755】ステップS903において、ファイル定義ブロック (FDB) のファイルアクセス認証方式 (Acceptable Authentication Type) が公開鍵認証を必要とする設定である場合は、ステップS904において、アクセスコマンドに必要な認証レベルの認証としての公開鍵認証が済んでいるか否かを判定し、認証未了の場合は、コマンドに従った処理を実行せず、アクセスエラーをファイルアクセス装置に送信 (S908) する。認証の終了または未了は、相互認証時に設定される認証テーブル (図51参照) に基づいて判定される。

【0756】ステップS903において、ファイル定義ブロック (FDB) のファイルアクセス認証方式 (Acceptable Authentication Type) が公開鍵認証を必要とする設定であり、ステップS904において、公開鍵認証が済んでいると判定された場合、あるいは、ファイル定義ブロック (FDB) のファイルアクセス認証方式 (Acceptable Authentication Type) が公開鍵認証を必要としない設定である場合は、次に、ステップS905において、デバイスのメモリ内の対応パーティションのファイル定義ブロック (FDB) (図24参照) に記録されたファイルアクセス検証方式 (Acceptable Verification Type :特定のファイル (File) アクセスを行う際に、指定する検証方式) を参照し、アクセス対象のファイルに対するアクセスコマンドの実行に必要な検証レベル (公開鍵方式の検証を必要とするか) を確認する。

【0757】ステップS905において、ファイル定義ブロック (FDB) のファイルアクセス検証方式 (Acceptable Verification Type) が公開鍵方式のチケット検証を必要とする設定である場合は、ステップS906において、アクセスコマンドに必要な検証レベルの検証としての公開鍵方式のチケット検証が済んでいるか否かを判定し、検証未了の場合は、コマンドに従った処理を実行せず、アクセスエラーをファイルアクセス装置に送信 (S908) する。

【0758】ステップS905において、ファイル定義ブロック (FDB) のファイルアクセス検証方式 (Acceptable Verification Type) が公開鍵方式のチケット検証を必要としない設定である場合は、ステップS907において、ファイルアクセス装置から受信したアクセスコマンドの処理を実行

し、結果をファイルアクセス装置に送信する。

【0759】アクセスコマンド結果を受信 (S892) したファイルアクセス装置は、さらに他のファイルアクセスを実行するか否かを判定 (S893) し、他のファイルアクセスを実行する場合は、ステップS891以下を繰り返し実行し、他にファイルアクセスを実行しない場合は処理を終了する。

【0760】次に、図87を用いて複数ファイルに対してアクセスを行なう場合の処理 (Combination) について説明する。図87のフローにおいて、左側がファイルアクセス装置、右側がデバイス (図5参照) の処理を示す。なお、ファイルアクセス処理において、ファイルアクセス装置とデバイス間におけるデータ転送の際には、相互認証処理において取得したセッション鍵 *Keys*、あるいは複数のセッション鍵から演算または選択されたセッション鍵を用いて暗号化され、また改竄チェック用のMACの生成、検証処理が実行される。

【0761】ファイルアクセス装置は、ステップS911において、アクセスコマンドをデバイスに送信する。このコマンドは、アクセス対象のファイルID (ソース)、ターゲットファイルID、アクセスモードを指定したコマンドであり、例えば先に図84を用いて説明したような、ソースファイルID [0x0002] に対して、ターゲットファイルID [0x0001] の鍵による内部暗号化処理の実行を指定するコマンド [Internal Encryption Command (0x0001, 0x0002)] などである。

【0762】デバイスは、ファイルアクセス装置からのコマンドを受信 (S921) すると、ファイルオープンテーブルのターゲットファイルIDのエントリにアクセスコマンドの許可があるか否かを判定 (S922) する。ファイルオープンテーブルのターゲットファイルIDのエントリにアクセスコマンドの許可が存在しない場合は、コマンドに従った処理を実行せず、アクセスエラーをファイルアクセス装置に送信 (S934) する。

【0763】ファイルオープンテーブルのターゲットファイルIDのエントリにアクセスコマンドの許可が存在した場合は、ステップS923において、デバイスのメモリ内の対応パーティションのファイル定義ブロック

(FDB) (図24参照) に記録されたファイルアクセス認証方式 (Acceptable Authentication Type :特定のファイル (File) アクセスを行う際に、指定する認証方式) を参照し、アクセス対象のターゲットファイルに対するアクセスコマンドの実行に必要な認証レベル (公開鍵認証を必要とするか) を確認する。

【0764】ステップS923において、アクセス対象のターゲットファイルに対して設定されたファイル定義ブロック (FDB) のファイルアクセス認証方式 (Acceptable Authentication Type) が公開鍵認証を必要とする設定である場合は、ステップS924において、アク

セスコマンドに必要な認証レベルの認証としての公開鍵認証が済んでいるか否かを判定し、認証未了の場合は、コマンドに従った処理を実行せず、アクセスエラーをファイルアクセス装置に送信（S 9 3 4）する。認証の終了または未了は、相互認証時に設定される認証テーブル（図 5 1 参照）に基づいて判定される。

【0 7 6 5】ステップ S 9 2 3 において、アクセス対象のターゲットファイルに対して設定されたファイル定義ブロック（FDB）のファイルアクセス認証方式（Acceptable Authentication Type）が公開鍵認証を必要とする設定であり、ステップ S 9 2 4 において、公開鍵認証が済んでいると判定された場合、あるいは、ファイル定義ブロック（FDB）のファイルアクセス認証方式（Acceptable Authentication Type）が公開鍵認証を必要としない設定である場合は、次に、ステップ S 9 2 5 において、デバイスのメモリ内の対応パーティションのファイル定義ブロック（FDB）（図 2 4 参照）に記録されたファイルアクセス検証方式（Acceptable Verification Type：特定のファイル（File）アクセスを行う際に、指定する検証方式）を参照し、アクセス対象のターゲットファイルに対するアクセスコマンドの実行に必要な検証レベル（公開鍵方式の検証を必要とするか）を確認する。

【0 7 6 6】ステップ S 9 2 5 において、アクセス対象のターゲットファイルに対して設定されたファイル定義ブロック（FDB）のファイルアクセス検証方式（Acceptable Verification Type）が公開鍵方式のチケット検証を必要とする設定である場合は、ステップ S 9 2 6 において、アクセスコマンドに必要な検証レベルの検証としての公開鍵方式のチケット検証が済んでいるか否かを判定し、検証未了の場合は、コマンドに従った処理を実行せず、アクセスエラーをファイルアクセス装置に送信（S 9 3 4）する。

【0 7 6 7】ステップ S 9 2 5 において、アクセス対象のターゲットファイルに対して設定されたファイル定義ブロック（FDB）のファイルアクセス検証方式（Acceptable Verification Type）が公開鍵方式のチケット検証を必要とする設定であり、ステップ S 9 2 6 において、公開鍵方式のチケット検証が済んでいると判定された場合、あるいは、ファイル定義ブロック（FDB）のファイルアクセス検証方式（Acceptable Verification Type）が公開鍵方式のチケット検証を必要としない設定である場合は、次に、ステップ S 9 2 7 において、アクセスコマンドに含まれるターゲットファイル ID によって指定されるファイルのアクセス方法（Read/Write）をコマンドに基づいて確認する。

【0 7 6 8】デバイスは、ファイルアクセス装置からのコマンドに含まれるソースファイル ID によって指定されるファイルがアクセスコマンドに含まれるアクセス方法（Read/Write）に対してオープンしているか否かを判

定（S 9 2 8）する。ファイルオープンテーブルにコマンド実行のためのアクセス方法（Read/Write）が存在しない場合は、コマンドに従った処理を実行せず、アクセスエラーをファイルアクセス装置に送信（S 9 3 4）する。

【0 7 6 9】ファイルオープンテーブルにコマンドに対応するアクセス方法（Read/Write）が存在した場合は、ステップ S 9 2 9 において、デバイスのメモリ内の対応パーティションのファイル定義ブロック（FDB）（図 2 4 参照）に記録されたファイルアクセス認証方式（Acceptable Authentication Type：特定のファイル（File）アクセスを行う際に、指定する認証方式）を参照し、アクセス対象のソースファイルに対するアクセスコマンドの実行に必要な認証レベル（公開鍵認証を必要とするか）を確認する。

【0 7 7 0】ステップ S 9 2 9 において、アクセス対象のソースファイルに対して設定されたファイル定義ブロック（FDB）のファイルアクセス認証方式（Acceptable Authentication Type）が公開鍵認証を必要とする設定である場合は、ステップ S 9 3 0 において、アクセスコマンドに必要な認証レベルの認証としての公開鍵認証が済んでいるか否かを判定し、認証未了の場合は、コマンドに従った処理を実行せず、アクセスエラーをファイルアクセス装置に送信（S 9 3 4）する。認証の終了または未了は、相互認証時に設定される認証テーブル（図 5 1 参照）に基づいて判定される。

【0 7 7 1】ステップ S 9 2 9 において、アクセス対象のソースファイルに対して設定されたファイル定義ブロック（FDB）のファイルアクセス認証方式（Acceptable Authentication Type）が公開鍵認証を必要とする設定であり、ステップ S 9 3 0 において、公開鍵認証が済んでいると判定された場合、あるいは、ファイル定義ブロック（FDB）のファイルアクセス認証方式（Acceptable Authentication Type）が公開鍵認証を必要としない設定である場合は、次に、ステップ S 9 3 1 において、デバイスのメモリ内の対応パーティションのファイル定義ブロック（FDB）（図 2 4 参照）に記録されたファイルアクセス検証方式（Acceptable Verification Type：特定のファイル（File）アクセスを行う際に、指定する検証方式）を参照し、アクセス対象のソースファイルに対するアクセスコマンドの実行に必要な検証レベル（公開鍵方式の検証を必要とするか）を確認する。

【0 7 7 2】ステップ S 9 3 1 において、アクセス対象のソースファイルに対して設定されたファイル定義ブロック（FDB）のファイルアクセス検証方式（Acceptable Verification Type）が公開鍵方式のチケット検証を必要とする設定である場合は、ステップ S 9 3 2 において、アクセスコマンドに必要な検証レベルの検証としての公開鍵方式のチケット検証が済んでいるか否かを判定し、検証未了の場合は、コマンドに従った処理を実行せ

ず、アクセスエラーをファイルアクセス装置に送信（S 9 3 4）する。

【0 7 7 3】ステップS 9 3 1において、アクセス対象のソースファイルに対して設定されたファイル定義ブロック（FDB）のファイルアクセス検証方式（Acceptable Verification Type）が公開鍵方式のチケット検証を必要とする設定であり、ステップS 9 3 2において、公開鍵方式のチケット検証が済んでいると判定された場合、あるいは、ファイル定義ブロック（FDB）のファイルアクセス検証方式（Acceptable Verification Type）が公開鍵方式のチケット検証を必要としない設定である場合は、ステップS 9 3 3において、ファイルアクセス装置から受信したアクセスコマンドの処理を実行し、結果をファイルアクセス装置に送信する。

【0 7 7 4】アクセスコマンド結果を受信（S 9 1 2）したファイルアクセス装置は、さらに他のファイルアクセスを実行するか否かを判定（S 9 1 3）し、他のファイルアクセスを実行する場合は、ステップS 9 1 1以下を繰り返し実行し、他にファイルアクセスを実行しない場合は処理を終了する。

【0 7 7 5】上述したファイルアクセス処理は、ファイル内にある1つのファイル構造によって指定されるデータが格納された場合の処理を想定して説明しているが、異なるファイル構造データを1つのファイル内に格納し、1つのファイルに対する1つのコマンドにより、上述の複数ファイルに対するシーケンシャル処理と同様の処理を実行する構成も可能である。

【0 7 7 6】図8 8に1つのファイルに対する1つのコマンドにより、1ファイル内のデータに対してシーケンシャル処理を実行する構成を説明する図を示す。

【0 7 7 7】ファイルは図に示すように、電子マネーファイルであり、金額データとしての[Purse]、利用ログデータとしての「Log」、データに対する暗号化または復号用の鍵データとしての[Key]から構成される。

【0 7 7 8】例えば、図8 8（a）に示すように、預け入れコマンド（Deposit Command）を規定し、ファイル内の金額データとしての[Purse]にX円を加算（S 9 4 1）し、さらに、ファイル内の利用ログデータとしての「Log」に[Purse]にX円を加算した記録を書き込む（S 9 4 2）という2つの処理を実行させる構成とすることが可能である。

【0 7 7 9】先に説明したファイルアクセスモード（図2 9参照）の入金系に対応する許容コマンド（図3 0参照）として、上述の預け入れコマンド（Deposit Command）を定義し、アクセス許可チケットのファイルアクセスモード（File Access Mode）に[入金系]を設定し、ファイルID（File ID）として、電子マネーを構成する複合ファイルを指定したアクセス許可チケット（SPT）を生成して、ファイルアクセス装置からデバイスに

対して送信した後、預け入れコマンド（Deposit Command）とともに、預け入れ金額データを送信することにより、図8 8（a）に示すようなデバイスにおいて1つのファイル内のデータに対するシーケンシャル処理を実行させることが可能となる。

【0 7 8 0】また、図8 8（b）に示すように、レシート生成コマンド（Make Receipt Command）を規定し、ファイル内の金額データとしての[Purse]からX円を減算（S 9 4 5）し、さらに、ファイル内の利用ログデータとしての「Log」に[Purse]からX円を減算した記録を書き込み（S 9 4 6）、さらに「Log」にデータに対する暗号化鍵データとしての[Key]を適用して署名をつけて送信する（S 9 4 7）という3ステップの処理を実行させる構成とすることも可能である。

【0 7 8 1】この場合はファイルアクセスモード（図2 9参照）の出金系に対応する許容コマンド（図3 0参照）として、上述のレシート生成コマンド（Make Receipt Command）を定義し、アクセス許可チケットのファイルアクセスモード（File Access Mode）に[出金系]を設定し、ファイルID（File ID）として、電子マネーを構成する複合ファイルを指定したアクセス許可チケット（SPT）を生成して、ファイルアクセス装置からデバイスに対して送信した後、レシート生成コマンド（Make Receipt Command）とともに、引き出し金額データを送信することにより、図8 8（b）に示すようなデバイスにおいて1つのファイル内のデータに対するシーケンシャル処理を実行させることが可能となる。

【0 7 8 2】このようにデバイスは、サービス許可チケット（SPT）に指定された処理ファイルが複合ファイルである場合、アクセス機器からの受領コマンドの処理対象ファイルを複合ファイル内から選択して処理を実行する。アクセス機器からのデータ処理コマンドが一連の複数の処理を含むシーケンス処理コマンドである場合は、デバイスは、シーケンス処理コマンドに含まれる各コマンドの処理対象ファイルを、サービス許可チケット（SPT）によって指定された複合ファイル内から順次選択して実行する。

【0 7 8 3】[B 4. 7. サービス許可チケット（SPT）を利用したアクセス処理各方式における処理手順] 上述したサービス許可チケット（SPT）を利用したアクセス処理ファイルの設定登録処理において、パーティションマネージャが管理し、サービス許可チケット（SPT）ユーザであるデバイスアクセス機器としてのリーダライタとデバイス間において、相互認証が実行され、サービス許可チケット（SPT）に基づくファイルアクセスがなされる。相互認証処理の態様は、公開鍵相互認証、共通鍵相互認証の2種類のいずれかであり、またチケット（SPT）の検証処理も公開鍵系の署名検証、共通鍵系のMAC検証の2種類のいずれかが実行されるこ

- とになる。すなわち処理態様としては大きく分けて、
- (A) 相互認証 (公開鍵)、チケット (SPT) 検証 (公開鍵)
- (B) 相互認証 (公開鍵)、チケット (SPT) 検証 (共通鍵)
- (C) 相互認証 (共通鍵)、チケット (SPT) 検証 (共通鍵)
- (D) 相互認証 (共通鍵)、チケット (SPT) 検証 (公開鍵)

の4態様がある。

【0784】これらの4態様についての処理を、認証局 (CA (PM))、パーティションマネージャ (PM)、SPTチケットユーザであるデバイスアクセス機器としてのリーダライタ、デバイス、各エンティティ間において実行されるデータ転送処理を中心として図を用いて簡潔に説明する。

【0785】(A) 相互認証 (公開鍵)、チケット (SPT) 検証 (公開鍵)

まず、相互認証処理に公開鍵方式を適用し、チケット (SPT) 検証に公開鍵方式を適用する場合の各エンティティ間のデータ転送について図89を用いて説明する。

【0786】図に示す番号順に各エンティティ間でデータ転送が実行される。以下、各番号に従って処理を説明する。

(1) パーティションマネージャ (PM) の公開鍵証明書 (Cert. PM) の発行、パーティションマネージャ (PM) の公開鍵証明書 (Cert. PM) は、パーティションマネージャ (PM) からの発行要求により、登録局 (RA) を介した証明書発行手続きによってパーティション対応認証局 (CA (PAR)) から発行される。なお、本構成は、パーティションマネージャがサービス許可チケット発行手段 (SPT Issuer) を兼ねる構成であり、サービス許可チケット発行手段 (SPT Issuer) の公開鍵証明書としてパーティションマネージャ (PM) の公開鍵証明書を使用する構成である。

【0787】(2) サービス許可チケットユーザ (SPT User) であるデバイスアクセス機器としてのリーダライタ (R/W) の公開鍵証明書 (Cert. RW) の発行、サービス許可チケットユーザ (SPT User: 具体的には、デバイスに対してチケットを送信するデバイスアクセス機器としてのリーダライタ (R/W)) の公開鍵証明書 (Cert. R/W) は、サービス許可チケットユーザ (SPT User) であるリーダライタ (R/W) からの発行要求により、登録局 (RA) を介した証明書発行手続きによってパーティション対応認証局 (CA (PAR)) によって発行される。なお、パーティションマネージャがサービス許可チケットユーザ (SPT User) を兼ねる構成も可能であり、その場合は、サービス許可チケットユーザ (SPT User) の公開鍵証明

書としてパーティションマネージャ (PM) の公開鍵証明書を使用可能である。

【0788】(3) サービス許可チケット (SPT) の生成処理

サービス許可チケット (SPT) は、パーティションマネージャの管理するサービス許可チケット発行手段 (SPT Ticket Issuer) により生成される。この場合、公開鍵方式の署名生成、検証を実行するため、サービス許可チケット発行手段 (SPT Ticket Issuer) の秘密鍵による署名 (Signature) が生成 (図12参照) されてSPTに付加される。

【0789】(4) SPTおよびサービス許可チケット発行手段 (SPT Ticket Issuer) としてのパーティションマネージャ公開鍵証明書 (Cert. PM) のサービス許可チケットユーザ (SPT User) であるデバイスアクセス機器としてのリーダライタ (R/W) に対する供給

パーティションマネージャの管理するサービス許可チケット発行手段 (SPT Ticket Issuer) により発行されたサービス許可チケット (SPT) は、サービス許可チケット発行手段 (SPT Ticket Issuer) としてのパーティションマネージャ公開鍵証明書 (Cert. PM) とともにサービス許可チケットユーザ (SPT User) であるデバイスアクセス機器としてのリーダライタ (R/W) に対して送信される。

【0790】(5) デバイスアクセス機器としてのリーダライタ (R/W) とデバイス間の相互認証  
サービス許可チケットユーザ (SPT User) であるリーダライタは、サービス許可チケット発行手段 (SPT Ticket Issuer) の発行したサービス許可チケット (SPT) に従ったファイルアクセスを実行しようとする対象のデバイスに対し、チケットユーザ (SPT User) としてのリーダライタ (R/W) の公開鍵証明書 (Cert. RW) をデバイスに送信し、公開鍵方式の相互認証 (図50参照) を実行する。

【0791】(6) SPTおよびサービス許可チケット発行手段 (SPT Ticket Issuer) としてのパーティションマネージャ公開鍵証明書 (Cert. PM) のデバイスに対する供給

デバイスアクセス機器としてのリーダライタ (R/W) とデバイス間の相互認証が成立すると、チケットユーザ (SPT User) としてのリーダライタ (R/W) は、デバイスに対してサービス許可チケット (SPT)、およびサービス許可チケット発行手段 (SPT Ticket Issuer) としてのパーティションマネージャ公開鍵証明書 (Cert. PM) を送信する。

【0792】デバイスは、受信したサービス許可チケット (SPT) について、(1) チケット発行者 (Ticket Issuer) としてのパーティションマネージャ公開鍵証明書 (Cert. PM) が改竄されたものでない正当な

公開鍵証明書 (CERT) であること、(2) チケット発行者 (Ticket Issuer) の公開鍵証明書 (CERT PM) のオプション領域に記録されたコードと、デバイス内のFDB (File Definition Block) に記録された (SPTIC) の一致、(3) チケット発行手段 (Ticket Issuer) がリポートされていないこと、(4) 受信チケット (SPT) の署名 (Signature) の検証によりチケットが改竄のないことの確認を実行し、さらに、SPTチケットに格納されたSPTユーザ (チケットユーザとしてのリーダライタ) とチケットユーザ (SPT User) の公開鍵証明書 (Cert. RW) の識別データ (DN) として記録された識別子またはカテゴリまたはシリアル (SN) 名 (DN) の一致を確認し、相互認証済みであることを確認することによりSPTユーザ (デバイスアクセス機器としてのリーダライタ) の検証 (図57、図58参照) を実行する。

【0793】 (7) ファイルアクセス

デバイスは、処理対象ファイルにサービス許可チケット (SPT) に記述されたルールに従ってアクセスを実行する。

【0794】 以上の処理によって、相互認証 (公開鍵)、チケット (SPT) 検証 (公開鍵) の各方式に従ったファイルアクセス処理が実行される。

【0795】 (B) 相互認証 (公開鍵)、チケット (SPT) 検証 (共通鍵)

次に、相互認証処理に公開鍵方式を適用し、チケット (SPT) 検証に共通鍵方式を適用する場合の各エンティティ間のデータ転送について図90を用いて説明する。

【0796】 図に示す番号順に各エンティティ間でデータ転送が実行される。以下、各番号に従って処理を説明する。

【0797】 (1) サービス許可チケットユーザ (SPT User) であるデバイスアクセス機器としてのリーダライタ (R/W) の公開鍵証明書 (Cert. RW) の発行、サービス許可チケットユーザ (SPT User: 具体的には、デバイスに対してチケットを送信するデバイスアクセス機器としてのリーダライタ (R/W)) の公開鍵証明書 (Cert. R/W) は、サービス許可チケットユーザ (SPT User) であるリーダライタ (R/W) からの発行要求により、登録局 (RA) を介した証明書発行手続きによってパーティション対応認証局 (CA (PAR)) によって発行される。なお、パーティションマネージャがサービス許可チケットユーザ (SPT User) を兼ねる構成も可能であり、その場合は、サービス許可チケットユーザ (SPT User) の公開鍵証明書としてパーティションマネージャ (PM) の公開鍵証明書を使用可能である。

【0798】 (2) サービス許可チケット (SPT) の生成処理

サービス許可チケット (SPT) は、パーティションマネージャの管理するサービス許可チケット発行手段 (SPT Ticket Issuer) により生成される。この場合、共通鍵方式の検証値としてMAC (Message Authentication Code) (図59参照) がSPTに付加される。

【0799】 (3) SPTのサービス許可チケットユーザ (SPT User) であるデバイスアクセス機器としてのリーダライタ (R/W) に対する供給  
パーティションマネージャの管理するサービス許可チケット発行手段 (SPT Ticket Issuer) により発行されたサービス許可チケット (SPT) は、サービス許可チケットユーザ (SPT User) としてのリーダライタ (R/W) に対して送信される。

【0800】 (4) リーダライタ (R/W) とデバイス間の相互認証

サービス許可チケットユーザ (SPT User) であるデバイスアクセス機器としてのリーダライタは、サービス許可チケット発行手段 (SPT Ticket Issuer) の発行したサービス許可チケット (SPT) に従ったファイルアクセスを実行しようとする対象のデバイスに対し、チケットユーザ (SPT User) としてのリーダライタ (R/W) の公開鍵証明書 (Cert. RW) をデバイスに送信し、公開鍵方式の相互認証 (図50参照) を実行する。パーティションマネージャ (PM) の公開鍵証明書を使用可能である。

【0801】 (5) SPTのデバイスに対する供給  
デバイスアクセス機器としてのリーダライタ (R/W) とデバイス間の相互認証が成立すると、サービス許可チケットユーザ (SPT User) であるリーダライタは、デバイスに対してサービス許可チケット (SPT) を送信する。デバイスは、受信したサービス許可チケット (SPT) についてMAC検証処理を実行し、SPT発行者 (SPT Issuer) の検証、さらに、SPTチケットに格納されたSPTユーザ (チケットユーザとしてのリーダライタ) とチケットユーザ (SPT User) の公開鍵証明書 (Cert. RW) の識別データ (DN) として記録された識別子またはカテゴリまたはシリアル (SN) 名 (DN) の一致を確認し、相互認証済みであることを確認することによりSPTユーザ (デバイスアクセス機器としてのリーダライタ) の検証 (図57、図58参照) を実行する。

【0802】 (6) ファイルアクセス

デバイスは、処理対象ファイルにサービス許可チケット (SPT) に記述されたルールに従ってアクセスを実行する。

【0803】 以上の処理によって、相互認証 (公開鍵)、チケット (SPT) 検証 (共通鍵) の各方式に従ったファイルアクセス処理が実行される。

【0804】 (C) 相互認証 (共通鍵)、チケット (SPT) 検証 (共通鍵)

次に、相互認証処理に共通鍵方式を適用し、チケット（SPT）検証に共通鍵方式を適用する場合の各エンティティ間のデータ転送について図91を用いて説明する。

【0805】図に示す番号順に各エンティティ間でデータ転送が実行される。以下、各番号に従って処理を説明する。

【0806】（1）サービス許可チケット（SPT）の生成処理

サービス許可チケット（SPT）は、パーティションマネージャの管理するサービス許可チケット発行手段（SPT Ticket Issuer）により生成される。この場合、共通鍵方式の検証値としてMAC（Message Authentication Code）（図59参照）がSPTに付加される。

【0807】（2）SPTのサービス許可チケットユーザ（SPT User）に対する供給  
パーティションマネージャの管理するサービス許可チケット発行手段（SPT Ticket Issuer）により発行されたサービス許可チケット（SPT）は、サービス許可チケットユーザ（SPT User）であるデバイスアクセス機器としてのリーダライタに対して送信される。

【0808】（3）デバイスアクセス機器としてのリーダライタ（R/W）とデバイス間の相互認証  
サービス許可チケットユーザ（SPT User）であるデバイスアクセス機器としてのリーダライタ（R/W）は、サービス許可チケット発行手段（SPT Ticket Issuer）の発行したサービス許可チケット（SPT）に従ったファイルを生成しようとする対象のデバイスとの間で、共通鍵方式の相互認証（図53、図54参照）を実行する。

【0809】（4）SPTのデバイスに対する供給  
デバイスアクセス機器としてのリーダライタ（R/W）とデバイス間の相互認証が成立すると、サービス許可チケットユーザ（SPT User）であるリーダライタは、デバイスに対してサービス許可チケット（SPT）を送信する。デバイスは、受信したサービス許可チケット（SPT）についてMAC検証処理を実行し、SPT発行者（SPT Issuer）の検証、さらに、SPTチケットに格納されたSPTユーザ（チケットユーザとしてのリーダライタ）とチケットユーザ（SPT User）の識別子の一致を確認し、相互認証済みであることを確認することによりSPTユーザ（デバイスアクセス機器としてのリーダライタ）の検証（図57、図58参照）を実行する。

【0810】（5）ファイルアクセス  
デバイスは、処理対象ファイルにサービス許可チケット（SPT）に記述されたルールに従ってアクセスを実行する。

【0811】以上の処理によって、相互認証（共通鍵）、チケット（SPT）検証（共通鍵）の各方式に従

ったファイルアクセス処理が実行される。

【0812】（D）相互認証（共通鍵）、チケット（SPT）検証（公開鍵）

次に、相互認証処理に共通鍵方式を適用し、チケット（SPT）検証に公開鍵方式を適用する場合の各エンティティ間のデータ転送について図92を用いて説明する。

【0813】図に示す番号順に各エンティティ間でデータ転送が実行される。以下、各番号に従って処理を説明する。

（1）パーティションマネージャ（PM）の公開鍵証明書（Cert. PM）の発行、パーティションマネージャ（PM）の公開鍵証明書（Cert. PM）は、パーティションマネージャ（PM）からの発行要求により、登録局（RA）を介した証明書発行手続きによってパーティション対応認証局（CA（PAR））から発行される。なお、本構成は、パーティションマネージャがサービス許可チケット発行手段（SPT Issuer）を兼ねる構成であり、サービス許可チケット発行手段（SPT Issuer）の公開鍵証明書としてパーティションマネージャ（PM）の公開鍵証明書を使用する構成である。

【0814】（2）サービス許可チケット（SPT）の生成処理

サービス許可チケット（SPT）は、パーティションマネージャの管理するサービス許可チケット発行手段（SPT Ticket Issuer）により生成される。この場合、公開鍵方式の署名生成、検証を実行するため、サービス許可チケット発行手段（SPT Ticket Issuer）の秘密鍵による署名（Signature）が生成（図12参照）されてSPTに付加される。

【0815】（3）SPTおよびサービス許可チケット発行手段（SPT Ticket Issuer）としてのパーティションマネージャ公開鍵証明書（Cert. PM）のサービス許可チケットユーザ（SPT User）であるデバイスアクセス機器としてのリーダライタ（R/W）に対する供給

パーティションマネージャの管理するサービス許可チケット発行手段（SPT Ticket Issuer）により発行されたサービス許可チケット（SPT）は、サービス許可チケット発行手段（SPT Ticket Issuer）としてのパーティションマネージャ公開鍵証明書（Cert. PM）とともにサービス許可チケットユーザ（SPT User）すなわち、デバイスに対してチケットを送信する機器（ex. デバイスアクセス機器としてのリーダライタ）に対して送信される。

【0816】（4）デバイスアクセス機器としてのリーダライタ（R/W）とデバイス間の相互認証  
サービス許可チケットユーザ（SPT User）であるデバイスアクセス機器としてのリーダライタは、サービス許可チケット発行手段（SPT Ticket Issuer）の発行



したサービス許可チケット (SPT) に従ったファイルアクセスを実行しようとする対象のデバイスとの間で、共通鍵方式の相互認証 (図53、図54参照) を実行する。

【0817】(5) SPTおよびサービス許可チケット発行手段 (SPT Ticket Issuer) としてのパーティションマネージャ公開鍵証明書 (Cert. PM) のデバイスに対する供給

リーダライタ (R/W) とデバイス間の相互認証が成立すると、サービス許可チケットユーザ (SPT User) であるデバイスアクセス機器としてのリーダライタは、デバイスに対してサービス許可チケット (SPT)、およびサービス許可チケット発行手段 (SPT Ticket Issuer) としてのパーティションマネージャ公開鍵証明書 (Cert. PM) を送信する。

【0818】デバイスは、受信したサービス許可チケット (SPT) について、(1) チケット発行者 (Ticket Issuer) としてのパーティションマネージャ公開鍵証明書 (Cert. PM) が改竄されたものでない正当な公開鍵証明書 (CERT) であること、(2) チケット発行者 (Ticket Issuer) としてのパーティションマネージャ公開鍵証明書 (Cert. PM) のオプション領域に記録されたコードと、デバイス内のFDB (File Definition Block) に記録されたチケット発行手段コード (SPTIC) の一致、(3) チケット発行手段 (Ticket Issuer) がリボークされていないこと、(4) 受信チケット (SPT) の署名 (Signature) の検証によりチケットが改竄のないことの確認を実行し、さらに、SPTチケットに格納されたSPTユーザ (チケットユーザとしてのリーダライタ) とチケットユーザ (SPT User) の識別子の一致を確認し、相互認証済みであることを確認することによりSPTユーザ (リーダライタ) の検証 (図57、図58参照) を実行する。

【0819】(6) ファイルアクセス

デバイスは、処理対象ファイルにサービス許可チケット (SPT) に記述されたルールに従ってアクセスを実行する。

【0820】以上の処理によって、相互認証 (共通鍵)、チケット (SPT) 検証 (公開鍵) の各方式に従ったファイルアクセス処理が実行される。

【0821】[B5. データアップデートチケット (DUT) を利用したデバイスのデータ更新処理] 次に、データアップデートチケット (DUT: Data Update Ticket) を利用したデバイスのデータ更新処理について説明する。データアップデートチケット (DUT: Data Update Ticket) は、デバイスに格納された様々なデータの更新処理を実行する際に適用されるアクセスコントロールチケットである。正当なデータアップデートチケット (DUT) 発行手段 (Ticket Issuer) の発行したDUTを用い、DUTに記録された手続きに従ってチケット

ユーザ (ex. デバイスアクセス機器としてのリーダライタ) によりデバイスにアクセスすることで、DUTに記録された制限内でデータ処理を実行することができる。

【0822】なお、前述したように、データアップデートチケット (DUT: Data Update Ticket) は、デバイスマネージャの管理するデータ項目の更新処理を実行するために適用されるチケットDUT (DEV) と、パーティションマネージャの管理するパーティション内のデータ項目の更新処理を実行するために適用されるチケットDUT (PAR) (図32参照) がある。

【0823】デバイスに格納したデータにデータアップデートチケット (DUT) を適用してデータ更新を実行する処理について説明する。図93以下のフロー他の図面を参照して説明する。なお、データ更新処理には、デバイスとデータ更新を実行するデバイスアクセス機器としてのリーダライタ間における相互認証処理 (デバイス認証またはパーティション認証)、データアップデートチケット (DUT: Data Update Ticket) の正当性検証処理が含まれる。

【0824】図93に示すデータ更新処理フローについて説明する。図93において、左側がデータ更新装置、右側がデバイス (図5参照) の処理を示す。なお、データ更新装置は、デバイスに対するデータ読み取り書き込み処理可能な装置 (ex. デバイスアクセス機器としてのリーダライタ、PC) であり、図10のデバイスアクセス機器としてのリーダライタに相当する。まず、図93を用いて、データ更新処理の概要を説明し、その後、当処理に含まれるデータ更新操作の詳細を図94のフローを用いて説明する。

【0825】まず、図93のステップS951とS960において、データ更新装置とデバイス間での相互認証処理が実行される。データ送受信を実行する2つの手段間では、相互に相手が正しいデータ通信者であるか否かを確認して、その後に必要なデータ転送を行なうことが行われる。相手が正しいデータ通信者であるか否かの確認処理が相互認証処理である。相互認証処理時にセッション鍵の生成を実行して、生成したセッション鍵を共有鍵として暗号化処理を実行してデータ送信を行なう。

【0826】相互認証処理については、先のパーティション生成、削除処理の欄で説明したと同様の処理であり、デバイス認証またはパーティション認証のいずれかが実行される。それぞれについて共通鍵方式認証、あるいは公開鍵方式認証処理のいずれかが適用される。この相互認証処理は、前述の図48～図56を用いて説明したと同様の処理であるので説明を省略する。

【0827】なお、相互認証処理として実行すべき処理は、適用するデータアップデートチケット (DUT)

(図32参照) の\* Authentication Type: デバイス (Device) の相互認証のタイプ (公開鍵認証、または、共

通鍵認証、または、いずれでも可 (Any) ) によって決定される。

【0828】認証処理に失敗した場合 (S952, S961でNo) は、相互が正当な機器、デバイスであることの確認がとれないことを示し、以下の処理は実行されずエラーとして処理は終了する。

【0829】認証処理に成功すると、データ更新装置は、デバイスに対してデータアップデートチケット (DUT: Data Update Ticket) を送信する。データアップデートチケット (DUT) は、デバイスマネージャまたはパーティションマネージャの管理下のデータアップデートチケット (DUT) 発行手段 (DUT Issuer) により発行されるチケットである。データアップデートチケット (DUT) は、デバイスに対するアクセス制御チケットであり、先に説明した図32のデータフォーマット構成を持つチケットである。

【0830】なお、データアップデートチケット (DUT) を、チケットユーザに対して送信する際には、公開鍵方式の場合、データアップデートチケット (DUT) 発行手段 (DUT Issuer) の公開鍵証明書 (CERT\_DUTI) も一緒に送信する。DUT発行手段の公開鍵証明書 (CERT\_DUTI) の属性 (Attribute) は、デバイス内のDKDB (PUB) (Device Key Definition Block) に記録されたチケット発行手段コード (DUTIC\_DEV) やPKDB (PUB) (Partition Key Definition Block) に記録されたチケット発行手段コード (DUTIC\_PAR) の識別子 (DUTIC) と一致する。

【0831】データアップデートチケット (DUT) を受信 (S962) したデバイスは、受信したチケット (DUT) の正当性と利用者チェック処理を実行 (S963) する。チケットの正当性の検証処理は、共通鍵方式によるMAC検証、あるいは公開鍵方式による署名検証処理のいずれかを適用して実行される。利用者チェックは、チケットを送信してきた機器 (チケット利用者) の正当性をチェックする処理であり、相互認証が成立済みであり、認証相手の識別データと、チケットに記録されているチケットユーザ識別子 (図32参照) との一致等を検証する処理として実行される。これらの処理は、先のパーティション登録チケット (PRT) の適用処理についての説明中、図57～図59を用いて説明したと同様の処理であるので説明を省略する。

【0832】デバイスにおいて、受信チケット (DUT) の正当性と利用者チェック処理の結果、チケットおよび利用者の正当なことが確認できなかった場合 (S964でNo) は、データアップデートチケット (DUT) 受理エラーをデータ更新装置に通知 (S968) する。チケットおよび利用者の正当なことが確認できた場合 (S964でYes) は、受信したデータアップデートチケット (DUT) に記述されたルールに従いデバイス内のメモリ部に格納されたデータ (図33参照) の更

新処理を実行する。この処理の詳細については、別フローを用いて後段で詳述する。

【0833】データアップデートチケット (DUT) の記述に従って、データの更新処理に成功 (S966でYes) すると、DUT受理成功をデータ更新装置に通知 (S967) する。一方、データの更新処理に失敗 (S966でNo) した場合は、DUT受理エラーをデータ更新装置に通知 (S968) する。

【0834】データ更新装置は、DUT受理結果を受信 (S954) し、DUT処理結果を判定し、DUT受理結果がエラーである場合 (S955でNo) は、エラーとして処理を終了し、DUT受理結果が成功 (S955でYes) である場合はセッションクリアコマンドの送受信 (S956, S969) を実行し、デバイス側に生成した認証テーブルを破棄 (S970) し、処理を終了する。認証テーブルは、ステップS951, S960の相互認証処理において生成されるテーブルであり、前述したパーティション登録チケット (PRT) の適用処理の項目において説明した構成、すなわち、図51の構成と同様のものである。

【0835】このようにデータアップデートチケット (DUT) を利用して、デバイス内に格納されたデータの更新処理が実行される。以下、当処理に含まれるデータ更新操作 (S965) について、図94を用いて説明する。

【0836】図94の処理フローは、データアップデートチケット (DUT) を受理したデバイスにおいて実行される処理であり、データアップデートチケット (DUT) を送信してきた機器との相互認証が成立し、チケットの検証にも成功した以後に実行される。

【0837】まず、ステップS971において、デバイスは、データアップデートチケット (DUT) の更新される古いデータのコード (Old Data Code) から更新対象データのバージョンを検索する。バージョンは、例えば更新対象がデバイスマネージャコード (DMC) であれば、デバイス管理情報ブロック (図15参照) にバージョンが記録され、また、パーティションマネージャコード (PMC) であれば、パーティション管理情報ブロック (図20参照) にバージョンが記録されている。また、パーティション登録チケット (PRT) 発行手段 (PRT Issuer) のバージョンはデバイス定義ブロック (図16参照) に含まれる。さらに、リボケーションリスト (IRL\_DEV, CRL\_DEV) などは、リボケーションリスト中にバージョン情報が含まれる。このように情報に応じてバージョン情報の格納先が決まっており、デバイスは、更新される古いデータのコード (Old Data Code) から更新対象データのバージョンを検索する。

【0838】次に、デバイスは、ステップS972において、データアップデートチケット (DUT) に記録さ

れたデータ更新をする時のバージョン条件 [Data Version Rule] を参照し、設定が [Any] であるか否かを判定する。

【0839】前述したように、データ更新をする時のバージョン条件 [Data Version Rule] は、Any、Exact、Older の3種類が存在する。Any はバージョン (Version) 条件に無関係でデータ更新が可能、Exact は、続く [Data Version Condition] に指定された値と同じ場合にデータ更新が可能、Older は、New Data Versionの方が新しい場合にのみデータ更新が可能となる。なお、バージョン条件 [Data Version Rule] がAny、または Older の場合は、[Data Version Condition] は使用しないかもしくは無視する。

【0840】データアップデートチケット (DUT) の [Data Version Rule] の設定が [Any] でない場合は、バージョン条件 [Data Version Rule] に従った処理を実行する。このステップがS973～S975である。

【0841】ステップS973では、データアップデートチケット (DUT) のバージョン条件 [Data Version Rule] を参照し、設定が [EXACT] であるか否かを判定する。[EXACT] は、[Data Version Condition] に指定された値と同じ場合にデータ更新が可能であることを示す。設定が [EXACT] である場合、ステップS974で、更新対象データ [Old Data] のバージョンがデータアップデートチケット (DUT) の [Data Version Condition] に記録されたバージョン値と一致するか否かを判定する。一致する場合にのみ次ステップに進み、一致しない場合は、更新処理を実行せずエラー終了とする。

【0842】ステップS973で、データアップデートチケット (DUT) のバージョン条件 [Data Version Rule] が [EXACT] でないと判定された場合は、設定は [Older] である。[Older] の設定は、更新対象データ [Old Data] のバージョンより、データアップデートチケット (DUT) の新規データ [New Data] のバージョンを示す [New Data Version] の方が新しい場合にのみ更新をする設定である。この [Older] の設定の場合、ステップS975において、更新対象データ [Old Data] のバージョンより、データアップデートチケット (DUT) の新規データ [New Data] のバージョンを示す [New Data Version] の方が新しいか否かを判定し、新しい場合にのみ次ステップに進み、一致しない場合は、更新処理を実行せずエラー終了とする。

【0843】次にデバイスは、ステップS976において、データアップデートチケット (DUT) の [Encrypted Flag] を検証する。[Encrypted Flag] は、更新されるデータが暗号化されているか否か(暗号化: Encrypted / 非暗号化: none)を示すデータである。[Encrypted

Flag] が更新対象データが非暗号化データであることを示している場合は、ステップS977において、データアップデートチケット (DUT) の新規データ [New Data] をデバイスのメモリ部に格納された更新対象旧データ [Old Data] に置き換える処理を実行し、処理終了とする。なお、更新対象データに対してバージョンが付加されている場合は、データアップデートチケット (DUT: Data Update Ticket) に格納されている更新するデータのバージョン (New Data Version) を、デバイス内の更新データに対応して設定されているバージョン格納領域に格納する処理を実行する。

【0844】また、ステップS976において、データアップデートチケット (DUT) の [Encrypted Flag] が、更新されるデータが暗号化されている(暗号化: Encrypted)ことを示していると判定された場合は、ステップS978において、データアップデートチケット (DUT) の [Ticket Type] を検証する。[Ticket Type] は、チケット (Ticket) の種別 (DUT (DEV) / DUT (PAR)) を示すデータである。DUT (DEV) は、データアップデートチケット (DUT) が、デバイスマネージャの管理するデータ項目の更新処理を実行するチケットであることを示し、DUT (PAR) は、パーティションマネージャの管理するパーティション内のデータ項目の更新処理を実行するために適用されるチケットであることを示してゐる。

【0845】チケットタイプ [Ticket Type] が、DUT (DEV) を示している場合、ステップS979～S982を実行し、DUT (PAR) を示している場合、ステップS983～S986を実行する。

【0846】チケットタイプ [Ticket Type] が、DUT (DEV) を示している場合、ステップS979において、データアップデートチケット (DUT (DEV)) に記述されたOld Data Code (更新される古いデータのコード) の示すデータがデバイス鍵領域 (図18参照) に格納されたKdut\_DEV1 (データアップデートチケット (DUT) のMAC検証用鍵) または、Kdut\_DEV2 (データ更新用暗号鍵) であるか否かを判定する。

【0847】データアップデートチケット (DUT (DEV)) に記述されたOld Data Code (更新される古いデータのコード) の示すデータがデバイス鍵領域 (図18参照) に格納されたKdut\_DEV1 (データアップデートチケット (DUT) のMAC検証用鍵)、Kdut\_DEV2 (データ更新用暗号鍵) である場合は、ステップS980において、デバイスのデバイス鍵領域 (図18参照) に格納されたKdut\_DEV4 (データ更新用暗号鍵) を用いて、データアップデートチケット (DUT (DEV)) に格納された新規データ [New Data] としてのKdut\_DEV1、Kdut\_DEV2を復号し、デバイスのデバイス鍵領域に格納されたKdut\_DEV1、Kdut\_DEV2に書きする。なお、データアップデートチケット (DUT (DEV)) に格納

されている更新するデータのバージョン (New Data Version) を、デバイス内の更新データに対応して設定されているバージョン格納領域、この場合は、デバイスのデバイス鍵領域 (図18参照) に格納する処理を併せて実行する。

【0848】次に、ステップS981において、デバイスのデバイス鍵領域 (図18参照) に格納されたKdut\_DEV1 (データアップデートチケット (DUT) のMAC検証用鍵) と、Kdut\_DEV3 (データアップデートチケット (DUT) のMAC検証用鍵) とのスワップ、すなわち入れ替え処理を行ない、また、Kdut\_DEV2 (データ更新用暗号鍵) と、Kdut\_DEV4 (データ更新用暗号鍵) とのスワップ、すなわち入れ替え処理を行なって処理を終了する。

【0849】なお、Kdut\_DEV1と、Kdut\_DEV3とのスワップ、および、Kdut\_DEV2と、Kdut\_DEV4とのスワップ処理によって、常にKdut\_DEV3 (データアップデートチケット (DUT) のMAC検証用鍵)、Kdut\_DEV4 (データ更新用暗号鍵) のペアがKdut\_DEV1 (データアップデートチケット (DUT) のMAC検証用鍵)、Kdut\_DEV2 (データ更新用暗号鍵) のペアよりも新しいバージョンのものに維持され、書き換え対象を、常にKdut\_DEV1、Kdut\_DEV2として設定した処理が可能となる。

【0850】なお、ステップS979において、データアップデートチケット (DUT (DEV)) に記述されたOld Data Code (更新される古いデータのコード) の示すデータがデバイス鍵領域 (図18参照) に格納されたKdut\_DEV1 (データアップデートチケット (DUT) のMAC検証用鍵)、Kdut\_DEV2 (データ更新用暗号鍵) でない場合は、ステップS982において、デバイスのデバイス鍵領域 (図18参照) に格納されたKdut\_DEV2 (データ更新用暗号鍵) を用いて、データアップデートチケット (DUT (DEV)) に格納された新規データ [New Data] を復号し、データアップデートチケット (DUT (DEV)) のOld Data code (更新される古いデータのコード) の示すエリアに上書きする。なお、更新対象データに対してバージョンが付加されている場合は、データアップデートチケット (DUT (DEV)) に格納されている更新するデータのバージョン (New Data Version) を、デバイス内の更新データに対応して設定されているバージョン格納領域に格納する処理を実行する。

【0851】一方、ステップS978において、チケットタイプ [Ticket Type] が、DUT (PAR) を示している場合、ステップS983～S986を実行する。

【0852】チケットタイプ [Ticket Type] が、DUT (PAR) を示している場合、ステップS983において、データアップデートチケット (DUT (PAR)) に記述されたOld Data Code (更新される古いデータのコード) の示すデータがパーティション鍵領域

(図23参照) に格納されたKdut\_PAR1 (データアップデートチケット (DUT) のMAC検証用鍵) または、Kdut\_PAR2 (データ更新用暗号鍵) であるか否かを判定する。

【0853】データアップデートチケット (DUT (PAR)) に記述されたOld Data Code (更新される古いデータのコード) の示すデータがパーティション鍵領域 (図23参照) に格納されたKdut\_PAR1 (データアップデートチケット (DUT) のMAC検証用鍵)、Kdut\_PAR2 (データ更新用暗号鍵) である場合は、ステップS984において、デバイスのパーティション鍵領域 (図23参照) に格納されたKdut\_PAR4 (データ更新用暗号鍵) を用いて、データアップデートチケット (DUT (PAR)) に格納された新規データ [New Data] としてのKdut\_PAR1、Kdut\_PAR2を復号し、デバイスのパーティション鍵領域に格納されたKdut\_PAR1、Kdut\_PAR2に上書きする。なお、データアップデートチケット (DUT (PAR)) に格納されている更新するデータのバージョン (New Data Version) を、デバイス内の更新データに対応して設定されているバージョン格納領域、この場合は、デバイスのパーティション鍵領域 (図23参照) に格納する処理を併せて実行する。

【0854】次に、ステップS985において、デバイスのパーティション鍵領域 (図23参照) に格納されたKdut\_PAR1 (データアップデートチケット (DUT) のMAC検証用鍵) と、Kdut\_PAR3 (データアップデートチケット (DUT) のMAC検証用鍵) とのスワップ、すなわち入れ替え処理を行ない、また、Kdut\_PAR2 (データ更新用暗号鍵) と、Kdut\_PAR4 (データ更新用暗号鍵) とのスワップ、すなわち入れ替え処理を行なって処理を終了する。

【0855】なお、Kdut\_PAR1と、Kdut\_PAR3とのスワップ、および、Kdut\_PAR2と、Kdut\_PAR4とのスワップ処理によって、常にKdut\_PAR3 (データアップデートチケット (DUT) のMAC検証用鍵)、Kdut\_PAR4 (データ更新用暗号鍵) のペアがKdut\_PAR1 (データアップデートチケット (DUT) のMAC検証用鍵)、Kdut\_PAR2 (データ更新用暗号鍵) のペアよりも新しいバージョンのものに維持され、書き換え対象を、常にKdut\_PAR1、Kdut\_PAR2として設定した処理が可能となる。

【0856】なお、ステップS983において、データアップデートチケット (DUT (PAR)) に記述されたOld Data Code (更新される古いデータのコード) の示すデータがデバイス鍵領域 (図18参照) に格納されたKdut\_DEV1 (データアップデートチケット (DUT) のMAC検証用鍵)、Kdut\_DEV2 (データ更新用暗号鍵) でない場合は、ステップS986において、デバイスのパーティション鍵領域 (図23参照) に格納されたKdut\_PAR2 (データ更新用暗号鍵) を用いて、データアップデートチケット (DUT (PAR)) に格納された

新規データ [NewData] を復号し、データアップデートチケット (DUT (PAR)) のOld DataCode (更新される古いデータのコード) の示すエリアに上書きする。なお、更新対象データに対してバージョンが付加されている場合は、データアップデートチケット (DUT (PAR)) に格納されている更新するデータのバージョン (New Data Version) を、デバイス内の更新データに対応して設定されているバージョン格納領域に格納する処理を実行する。

【0857】以上の処理がデバイスにおいて実行されるデータアップデートチケットに基づくデータ更新操作である。

【0858】上述したフローから理解されるように、更新対象データがデバイス鍵領域に格納されたKdut\_DEV1 (データアップデートチケット (DUT) のMAC検証用鍵)

Kdut\_DEV2 (データ更新用暗号鍵)

または、パーティション鍵領域に格納された

Kdut\_PAR1 (データアップデートチケット (DUT) のMAC検証用鍵)

Kdut\_PAR2 (データ更新用暗号鍵)

である場合には、他の更新処理と異なる処理を実行する。

【0859】これらのKdut\_DEV1 (データアップデートチケット (DUT) のMAC検証用鍵)、Kdut\_DEV2 (データ更新用暗号鍵)、Kdut\_PAR1 (データアップデートチケット (DUT) のMAC検証用鍵)、Kdut\_PAR2 (データ更新用暗号鍵) についての更新処理を簡潔にまとめた図を図95に示し、処理について説明する。図95の(1)～(3)の順に説明する。なお、処理は、Kdut\_DEV1, 2と、Kdut\_PAR1, 2とで同様のものであるもので、Kdut\_DEV1, 2を更新する場合について説明する。

【0860】(1) データアップデートチケット (DUT) に格納する新規データ [New Data] としてのKdut\_DEV1、Kdut\_DEV2をデバイスのデバイス鍵領域 (図18参照) に格納されたKdut\_DEV4 (データ更新用暗号鍵) を用いて暗号化した後、データアップデートチケット (DUT) に格納し、データアップデートチケット (DUT) をデバイスに送信する。このとき、Kdut\_DEV1、Kdut\_DEV2を更新できるチケット発行者はKdut\_DEV3、Kdut\_DEV4を知らなくてはならない。

【0861】(2) データアップデートチケット (DUT) を受信したデバイスは、デバイスのデバイス鍵領域に格納されたKdut\_DEV4 (データ更新用暗号鍵) を用いて、データアップデートチケット (DUT) の格納新規データ [New Data] としてのKdut\_DEV1、Kdut\_DEV2を復号し、デバイスのデバイス鍵領域に格納されたKdut\_DEV1、Kdut\_DEV2に上書きする。

【0862】(3) 次に、デバイスは、デバイスのデバイス鍵領域 (図18参照) に新規に格納されたKdut\_DEV

1 (データアップデートチケット (DUT) のMAC検証用鍵) と、以前に格納済みのKdut\_DEV3 (データアップデートチケット (DUT) のMAC検証用鍵) とのスワップ、すなわち入れ替え処理を行なう。さらに、新規に格納されたKdut\_DEV2 (データ更新用暗号鍵) と、以前に格納済みのKdut\_DEV4 (データ更新用暗号鍵) とのスワップ、すなわち入れ替え処理を行なう。

【0863】この、Kdut\_DEV1と、Kdut\_DEV3とのスワップ、および、Kdut\_DEV2と、Kdut\_DEV4とのスワップ処理によって、常にKdut\_DEV3 (データアップデートチケット (DUT) のMAC検証用鍵)、Kdut\_DEV4 (データ更新用暗号鍵) のペアがKdut\_DEV1 (データアップデートチケット (DUT) のMAC検証用鍵)、Kdut\_DEV2 (データ更新用暗号鍵) のペアよりも新しいバージョンのものに維持される。つまり、Kdut\_DEV1と、Kdut\_DEV2の鍵は常に使用される鍵で、Kdut\_DEV3と、Kdut\_DEV4は、非常時にKdut\_DEV1と、Kdut\_DEV2を更新するとともに、現在使用されているKdut\_DEV1と、Kdut\_DEV2の鍵に置き換えられるバックアップ用の鍵としての役割がある。

【0864】なお、Kdut\_DEV1 (データアップデートチケット (DUT) のMAC検証用鍵)、Kdut\_DEV2 (データ更新用暗号鍵) はペアとして使用され、また、Kdut\_DEV3 (データアップデートチケット (DUT) のMAC検証用鍵)、Kdut\_DEV4 (データ更新用暗号鍵) もペアとして使用される。

【0865】以上、特定の実施例を参照しながら、本発明について詳解してきた。しかしながら、本発明の要旨を逸脱しない範囲で当業者が該実施例の修正や代用を成し得ることは自明である。すなわち、例示という形態で本発明を開示してきたのであり、限定的に解釈されるべきではない。本発明の要旨を判断するためには、冒頭に記載した特許請求の範囲の欄を参照すべきである。

【0866】なお、明細書中において説明した一連の処理はハードウェア、またはソフトウェア、あるいは両者の複合構成によって実行することが可能である。ソフトウェアによる処理を実行する場合は、処理シーケンスを記録したプログラムを、専用のハードウェアに組み込まれたコンピュータ内のメモリにインストールして実行させるか、あるいは、各種処理が実行可能な汎用コンピュータにプログラムをインストールして実行させることが可能である。

【0867】例えば、プログラムは記録媒体としてのハードディスクやROM (Read Only Memory) に予め記録しておくことができる。あるいは、プログラムはフロッピーディスク、CD-ROM (Compact Disc Read Only Memory)、MO (Magnetooptical) ディスク、DVD (Digital Versatile Disc)、磁気ディスク、半導体メモリなどのリムーバブル記録媒体に、一時的あるいは永続的に格納 (記録) しておくことができる。このようなリムーバ

ブル記録媒体は、いわゆるパッケージソフトウェアとして提供することができる。

【0868】なお、プログラムは、上述したようなリムーバブル記録媒体からコンピュータにインストールする他、ダウンロードサイトから、コンピュータに無線転送したり、LAN(Local Area Network)、インターネットといったネットワークを介して、コンピュータに有線で転送し、コンピュータでは、そのようにして転送されてくるプログラムを受信し、内蔵するハードディスク等の記録媒体にインストールすることができる。

【0869】なお、明細書に記載された各種の処理は、記載に従って時系列に実行されるのみならず、処理を実行する装置の処理能力あるいは必要に応じて並列的あるいは個別に実行されてもよい。また、本明細書においてシステムとは、複数の装置の論理的集合構成であり、各構成の装置が同一筐体内にあるものには限らない。

【0870】

【発明の効果】上述したように、本発明のメモリアクセス制御システム、デバイス管理装置、パーティション管理装置、メモリ搭載デバイス、およびメモリアクセス制御方法、並びにプログラム記憶媒体によれば、複数のパーティションに分割されたメモリ領域のアクセスに対して、様々な種類のアクセス制御チケットを各デバイスまたはパーティション管理エンティティの管理の下に発行し、各チケットに記述されたルールに基づく処理をメモリ搭載デバイスにおいて実行する構成が可能となり、各パーティション内データの独立した管理構成が実現される。

【0871】さらに、本発明のメモリアクセス制御システム、デバイス管理装置、パーティション管理装置、メモリ搭載デバイス、およびメモリアクセス制御方法、並びにプログラム記憶媒体によれば、パーティション対応の認証、デバイス対象の認証を公開鍵、共通鍵のいずれか指定方式に従って実行することが可能なデバイスとし、様々な環境下においてデバイスおよびアクセス装置間のセキュアなデータ通信が実行可能となる。

【0872】さらに、本発明のメモリアクセス制御システム、デバイス管理装置、パーティション管理装置、メモリ搭載デバイス、およびメモリアクセス制御方法、並びにプログラム記憶媒体によれば、メモリ搭載デバイスのメモリ部は、データファイルを格納し、パーティションマネージャによって管理されるメモリ領域としての1以上のパーティション領域と、該メモリ搭載デバイスの管理者としてのデバイスマネージャによって管理されるデバイスマネージャ管理領域とを有し、メモリ部に対するアクセス制御チケットとして、デバイスマネージャの管理するアクセス制御チケット、またはパーティションマネージャの管理するアクセス制御チケットをアクセス機器から受領し、受領チケットの記述に応じて処理を実行する構成とし、実行すべき相互認証態様、アクセス制

御チケットの検証態様を指定し、各態様に従って処理を可能としたので様々な環境下においてデバイスおよびアクセス装置間のセキュアなデータ通信が実行可能となる。

【0873】さらに、本発明のメモリアクセス制御システム、デバイス管理装置、パーティション管理装置、メモリ搭載デバイス、およびメモリアクセス制御方法、並びにプログラム記憶媒体によれば、デバイスマネージャ、パーティションマネージャの管理の下、パーティション登録チケット(PRT)、ファイル登録チケット、サービス許可チケット(SPT)、データアップデートチケット(DUT)を発行し、それぞれ認証、チケット検証の成立を条件としてデバイスでの処理を実行する構成としたので、様々な処理態様に応じたサービスの提供、データ管理が各サービス主体の管理の下に実行可能となる。

【図面の簡単な説明】

【図1】本発明のシステム構成の概要を説明するシステム構成概略図(その1)である。

【図2】本発明のシステム構成の概要を説明するシステム構成概略図(その2)である。

【図3】本発明のシステム構成の具体例を説明するシステム構成概略図(その3)である。

【図4】本発明のシステムにおけるアクセス制御チケットの発行手段および利用手段との関係を説明する図である。

【図5】本発明のシステムにおけるメモリ部を有するデバイス構成を示す図である。

【図6】本発明のデバイスのメモリフォーマットを示す図である。

【図7】本発明のシステムにおけるデバイスマネージャ構成を示す図である。

【図8】本発明のシステムにおけるデバイスマネージャの制御手段構成を示す図である。

【図9】本発明のシステムにおけるパーティションマネージャ構成を示す図である。

【図10】本発明のシステムにおけるリーダライト(R/W)構成を示す図である。

【図11】本発明のシステムにおいて利用可能な公開鍵証明書のフォーマットを説明する図である。

【図12】本発明のシステムにおいて利用可能な公開鍵方式の署名生成処理フローを示す図である。

【図13】本発明のシステムにおいて利用可能な公開鍵方式の署名検証処理フローを示す図である。

【図14】本発明のデバイスにおけるメモリ部に格納されるデータ中の製造情報ブロックのデータ構成を示す図である。

【図15】本発明のデバイスにおけるメモリ部に格納されるデータ中のデバイス管理情報ブロックのデータ構成を示す図である。

【図16】本発明のデバイスにおけるメモリ部に格納されるデータ中の公開鍵系デバイス鍵定義ブロックのデータ構成を示す図である。

【図17】本発明のデバイスにおけるメモリ部に格納されるデータ中の共通鍵系デバイス鍵定義ブロックのデータ構成を示す図である。

【図18】本発明のデバイスにおけるメモリ部に格納されるデータ中のデバイス鍵領域のデータ構成を示す図である。

【図19】本発明のデバイスにおけるメモリ部に格納されるデータ中のパーティション定義ブロックのデータ構成を示す図である。

【図20】本発明のデバイスにおけるメモリ部に格納されるデータ中のパーティション管理情報ブロックのデータ構成を示す図である。

【図21】本発明のデバイスにおけるメモリ部に格納されるデータ中の公開鍵系パーティション鍵定義ブロックのデータ構成を示す図である。

【図22】本発明のデバイスにおけるメモリ部に格納されるデータ中の共通鍵系パーティション鍵定義ブロックのデータ構成を示す図である。

【図23】本発明のデバイスにおけるメモリ部に格納されるデータ中のパーティション鍵領域のデータ構成を示す図である。

【図24】本発明のデバイスにおけるメモリ部に格納されるデータ中のファイル定義ブロックのデータ構成を示す図である。

【図25】本発明のデバイスにおけるメモリ部に格納されるデータ中のファイルの構造のタイプについて説明する図である。

【図26】本発明のシステムにおいて適用されるアクセス制御チケットとしてのパーティション登録チケット(PRT)のフォーマットを示す図である。

【図27】本発明のシステムにおいて適用されるアクセス制御チケットとしてのファイル登録チケット(FRT)のフォーマットを示す図である。

【図28】本発明のシステムにおいて適用されるアクセス制御チケットとしてのサービス許可チケット(SPT)のフォーマット(例1)を示す図である。

【図29】本発明のシステムにおいて適用されるアクセス制御チケットとしてのサービス許可チケット(SPT)を利用したファイルアクセスのモードの種別を説明する図である。

【図30】本発明のシステムにおいて適用されるアクセス制御チケットとしてのサービス許可チケット(SPT)を利用したアクセス対象となるファイル構造を説明する図である。

【図31】本発明のシステムにおいて適用されるアクセス制御チケットとしてのサービス許可チケット(SPT)のフォーマット(例2)を示す図である。

【図32】本発明のシステムにおいて適用されるアクセス制御チケットとしてのデータアップデートチケット(DUT)のフォーマットを示す図である。

【図33】本発明のシステムにおいて適用されるアクセス制御チケットとしてのデータアップデートチケット(DUT)を利用した更新対象となるデータを説明する図である。

【図34】本発明のシステムにおけるデバイス利用までの処理概略を説明する図である。

【図35】本発明のシステムにおけるデバイス製造エンティティによるデバイスの初期登録処理フローを示す図である。

【図36】本発明のシステムにおけるデバイスマネージャによるデバイスの初期登録処理フロー(その1)を示す図である。

【図37】本発明のシステムにおけるデバイスマネージャによるデバイスの初期登録処理フロー(その2)を示す図である。

【図38】本発明のシステムにおけるデバイスマネージャによるデバイスの初期登録処理フロー(その3)を示す図である。

【図39】本発明のシステムにおけるデバイスマネージャによるデバイスの初期登録処理フロー(その4)を示す図である。

【図40】本発明のシステムにおけるデバイスマネージャによるデバイスの初期登録処理フロー(その5)を示す図である。

【図41】本発明のシステムにおけるデバイスマネージャによるデバイスの初期登録処理の後のデバイスの格納データを説明する図である。

【図42】本発明のシステムにおけるデバイスマネージャによる公開鍵証明書発行処理フロー(その1)を示す図である。

【図43】本発明のシステムにおけるデバイスマネージャによる公開鍵証明書発行処理フロー(その2)を示す図である。

【図44】本発明のシステムにおけるデバイスマネージャによる公開鍵証明書発行処理を説明する図である。

【図45】本発明のシステムにおけるデバイスマネージャによる公開鍵証明書発行処理を説明する図である。

【図46】本発明のシステムにおけるデバイスマネージャによる公開鍵証明書発行処理後のデバイスの格納データを説明する図である。

【図47】本発明のシステムにおけるデバイスに対するパーティション生成、削除処理フローを示す図である。

【図48】本発明のシステムにおけるデバイスとの相互認証処理について説明するフロー(その1)である。

【図49】本発明のシステムにおけるデバイスとの相互認証処理(デバイス認証)について説明するフロー(その2)である。

【図50】本発明のシステムにおけるデバイスとの公開鍵方式の相互認証処理について説明する図である。

【図51】本発明のシステムにおけるデバイスとの相互認証処理後にデバイスに生成される認証テーブルの構成を説明する図である。

【図52】本発明のシステムにおけるデバイスとの相互認証処理後にリーダライタに生成される認証テーブルの構成を説明する図である。

【図53】本発明のシステムにおけるデバイスとの共通鍵方式の相互認証処理について説明する図である。

【図54】本発明のシステムにおけるデバイスとの共通鍵方式の相互認証処理について説明する図である。

【図55】本発明のシステムにおけるデバイスとの相互認証処理（パーティション認証）について説明するフロー（その3）である。

【図56】本発明のシステムにおけるデバイスとの相互認証処理（パーティション認証）について説明するフロー（その4）である。

【図57】本発明のシステムにおけるチケットの正当性、利用者チェック処理について説明するフロー（その1）である。

【図58】本発明のシステムにおけるチケットの正当性、利用者チェック処理について説明するフロー（その2）である。

【図59】本発明のシステムにおけるチケットの正当性で適用可能なMAC生成方式について説明するフロー（その1）である。

【図60】本発明のシステムにおけるパーティションの作成、削除操作について説明するフロー（その1）である。

【図61】本発明のシステムにおけるパーティションの作成、削除操作について説明するフロー（その2）である。

【図62】本発明のシステムにおけるパーティションの初期登録処理について説明するフロー（その1）である。

【図63】本発明のシステムにおけるパーティションの初期登録処理について説明するフロー（その2）である。

【図64】本発明のシステムにおけるパーティションの初期登録処理について説明するフロー（その3）である。

【図65】本発明のシステムにおけるパーティションの初期登録処理後のデバイス格納データについて説明する図である。

【図66】本発明のシステムにおけるパーティションマネージャによる公開鍵証明書発行処理を説明する図（その1）である。

【図67】本発明のシステムにおけるパーティションマネージャによる公開鍵証明書発行処理を説明する図（そ

の2）である。

【図68】本発明のシステムにおけるパーティションマネージャによるパーティション生成処理において、公開鍵方式認証、公開鍵方式チケット検証を実行した場合の処理を説明する図である。

【図69】本発明のシステムにおけるパーティションマネージャによるパーティション生成処理において、公開鍵方式認証、共通鍵方式チケット検証を実行した場合の処理を説明する図である。

【図70】本発明のシステムにおけるパーティションマネージャによるパーティション生成処理において、共通鍵方式認証、共通鍵方式チケット検証を実行した場合の処理を説明する図である。

【図71】本発明のシステムにおけるパーティションマネージャによるパーティション生成処理において、共通鍵方式認証、公開鍵方式チケット検証を実行した場合の処理を説明する図である。

【図72】本発明のシステムにおけるファイル登録チケット（FRT）を適用したファイル生成消去処理について説明するフロー図である。

【図73】本発明のシステムにおけるファイル登録チケット（FRT）を適用したファイル生成削除操作について説明するフロー図である。

【図74】本発明のシステムにおけるファイル登録チケット（FRT）を適用したファイル生成後のデバイス格納データを説明する図である。

【図75】本発明のシステムにおけるファイル登録チケット（FRT）によるファイル生成処理において、公開鍵方式認証、公開鍵方式チケット検証を実行した場合の処理を説明する図である。

【図76】本発明のシステムにおけるファイル登録チケット（FRT）によるファイル生成処理において、公開鍵方式認証、共通鍵方式チケット検証を実行した場合の処理を説明する図である。

【図77】本発明のシステムにおけるファイル登録チケット（FRT）によるファイル生成処理において、共通鍵方式認証、共通鍵方式チケット検証を実行した場合の処理を説明する図である。

【図78】本発明のシステムにおけるファイル登録チケット（FRT）によるファイル生成処理において、共通鍵方式認証、公開鍵方式チケット検証を実行した場合の処理を説明する図である。

【図79】本発明のシステムにおけるサービス許可チケット（SPT）を適用したファイルアクセス処理フローを示す図である。

【図80】本発明のシステムにおけるサービス許可チケット（SPT）を適用したファイルオープン操作フローを示す図である。

【図81】本発明のシステムにおけるサービス許可チケット（SPT）を適用したファイルオープン操作により



生成されるファイルオープンテーブル構成を説明する図(例1)である。

【図82】本発明のシステムにおけるサービス許可チケット(SPT)を適用したファイルオープン操作により生成されるファイルオープンテーブル構成を説明する図(例2)である。

【図83】本発明のシステムにおけるサービス許可チケット(SPT)を適用したファイルアクセス処理例を説明する図(例1)である。

【図84】本発明のシステムにおけるサービス許可チケット(SPT)を適用したファイルアクセス処理例を説明する図(例2)である。

【図85】本発明のシステムにおける認証により生成されるセッション鍵の取扱について説明する図である。

【図86】本発明のシステムにおけるサービス許可チケット(SPT)を適用したファイルアクセス処理例を説明するフロー図(例1)である。

【図87】本発明のシステムにおけるサービス許可チケット(SPT)を適用したファイルアクセス処理例を説明するフロー図(例2)である。

【図88】本発明のシステムにおけるサービス許可チケット(SPT)を適用した複合ファイルのアクセス処理例を説明する図である。

【図89】本発明のシステムにおけるサービス許可チケット(SPT)によるファイルアクセス処理において、公開鍵方式認証、公開鍵方式チケット検証を実行した場合の処理を説明する図である。

【図90】本発明のシステムにおけるサービス許可チケット(SPT)による処理において、公開鍵方式認証、共通鍵方式チケット検証を実行した場合の処理を説明する図である。

【図91】本発明のシステムにおけるサービス許可チケット(SPT)による処理において、共通鍵方式認証、共通鍵方式チケット検証を実行した場合の処理を説明する図である。

【図92】本発明のシステムにおけるサービス許可チケット(SPT)による処理において、共通鍵方式認証、公開鍵方式チケット検証を実行した場合の処理を説明する図である。

【図93】本発明のシステムにおけるデータアップデートチケット(DUT)によるデータ更新処理フローを示す図である。

【図94】本発明のシステムにおけるデータアップデートチケット(DUT)によるデータ更新操作フローを示す図である。

【図95】本発明のシステムにおけるデータアップデートチケット(DUT)によるデータ更新処理例を説明する図である。

る図である。

【図96】従来のメモリ構造を示す図である。

【図97】従来のメモリ管理者、利用者の関係を説明する図である。

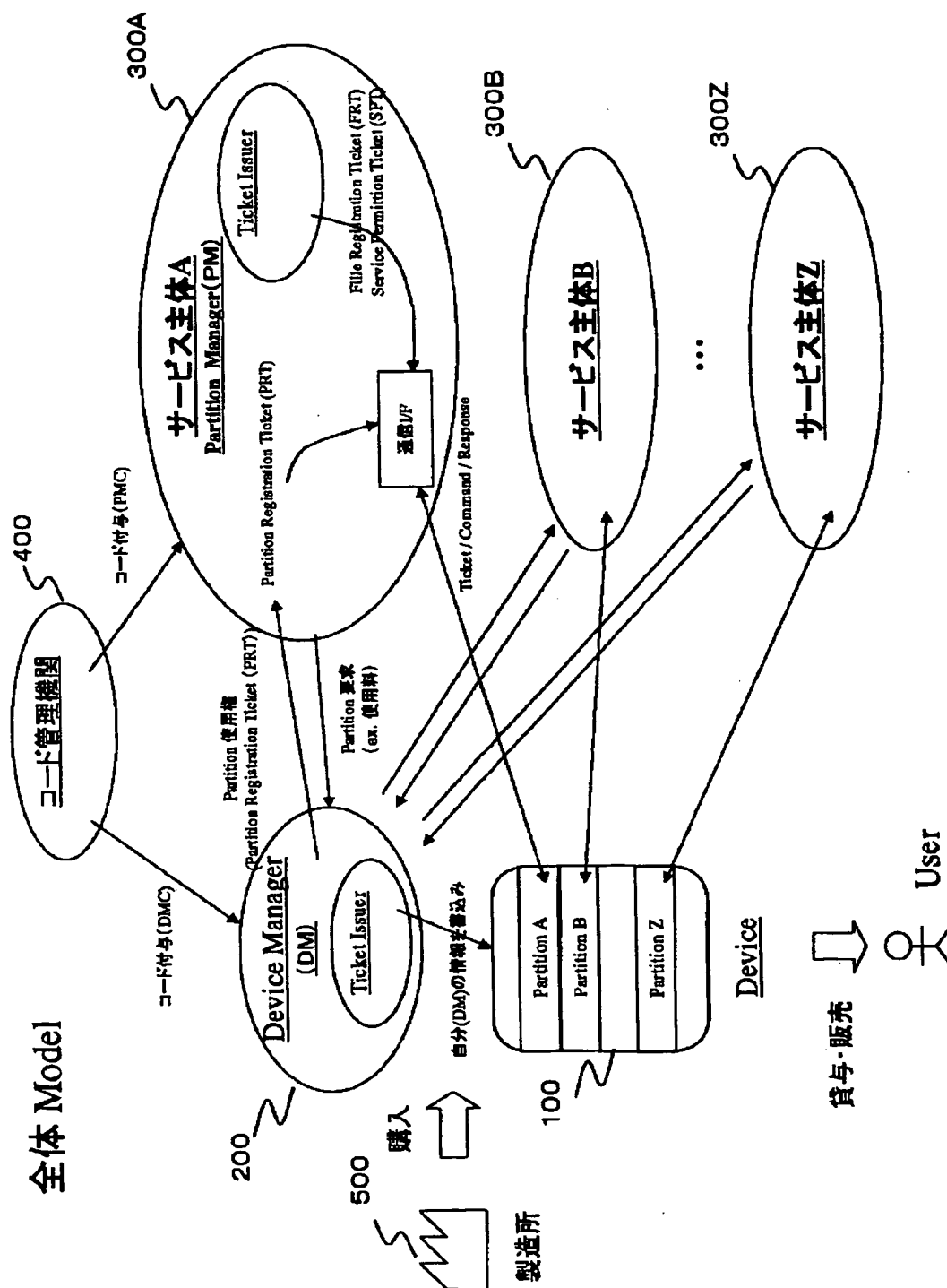
【図98】従来のメモリ領域確保処理について説明する図である。

【図99】従来のメモリアクセス方式について説明する図である。

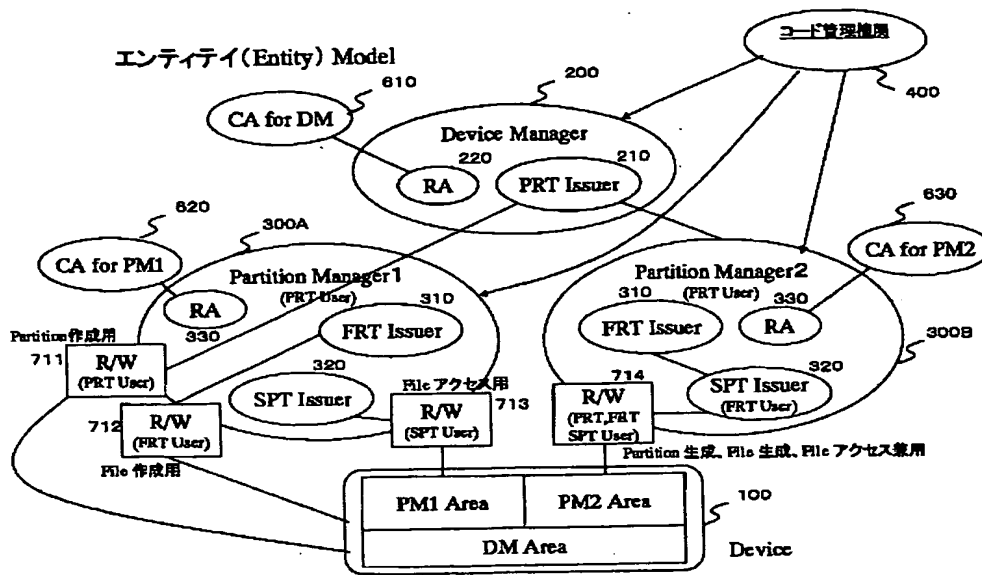
#### 【符号の説明】

100 デバイス  
200 デバイスマネージャ  
300 パーティションマネージャ  
400 コード管理機関  
500 デバイス製造エンティティ  
210 パーティション登録チケット(PRT)発行手段  
220 登録局(RA)  
310 ファイル登録チケット(FRT)発行手段  
320 サービス許可チケット(SPT)発行手段  
330 登録局(RA)  
610, 620 認証局(CA)  
711~714 リードライタ(R/W)  
721~722 チケットユーザ  
101 CPU  
102 通信IF  
103 ROM  
104 RAM  
105 暗号処理部  
106 メモリ部(EEPROM)  
211, 221 制御手段  
212, 222 データベース  
2111 制御部  
2112 ROM  
2113 RAM  
2114 表示部  
2115 入力部  
2116 HDD  
2117 ドライブ  
2118 通信I/F  
311, 321, 331 制御手段  
312, 322, 332 データベース  
701 CPU  
702 通信IF  
703 ROM  
704 RAM  
705 暗号処理部  
706 メモリ部(EEPROM)

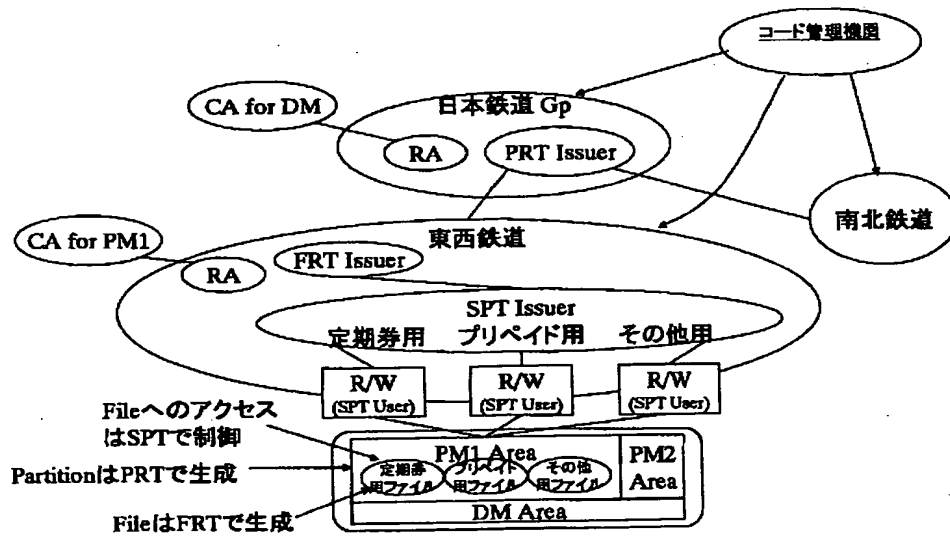
【図1】



【図2】



【図3】



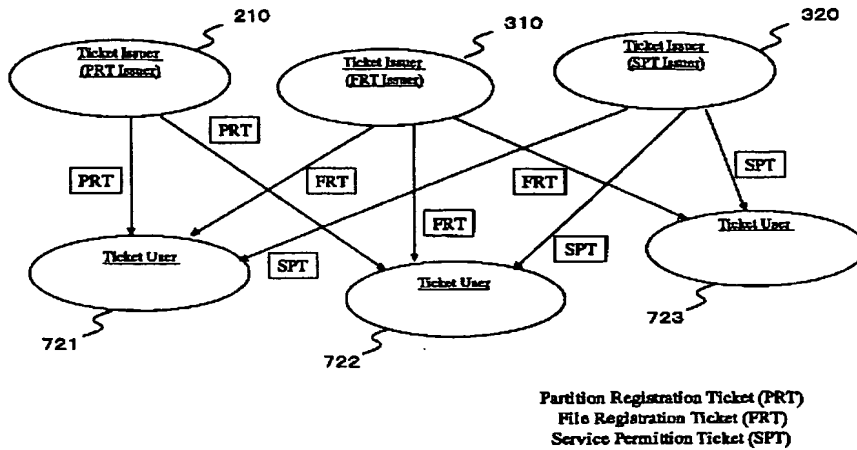
【図81】

File Open テーブル(1)

Group	File ID	File Access Mode
PMC1	0x0001	Enc, Dec
PMC1	0x0002	Read

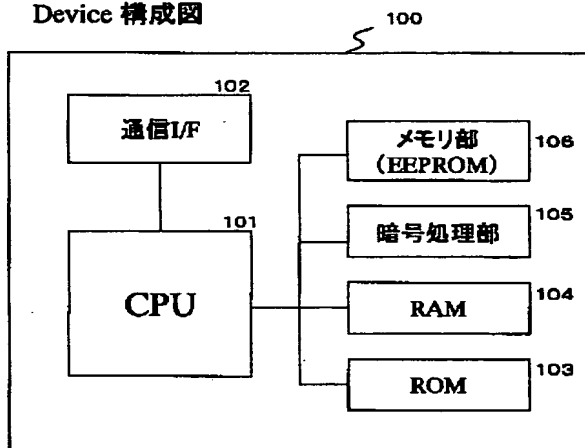
【図4】

## Ticket, Ticket Issuer &amp; Ticket User



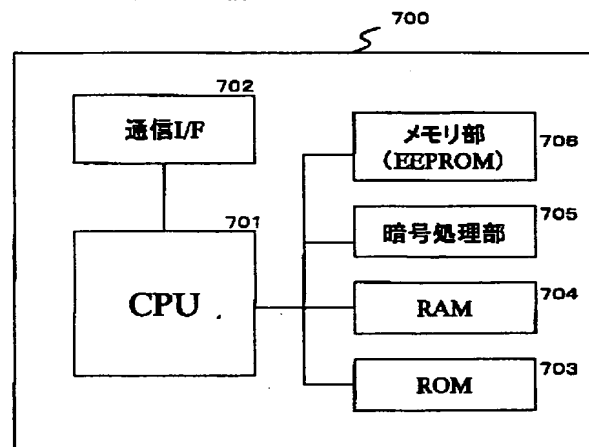
【図5】

## Device 構成図



【図10】

## Reader/Writer 構成図

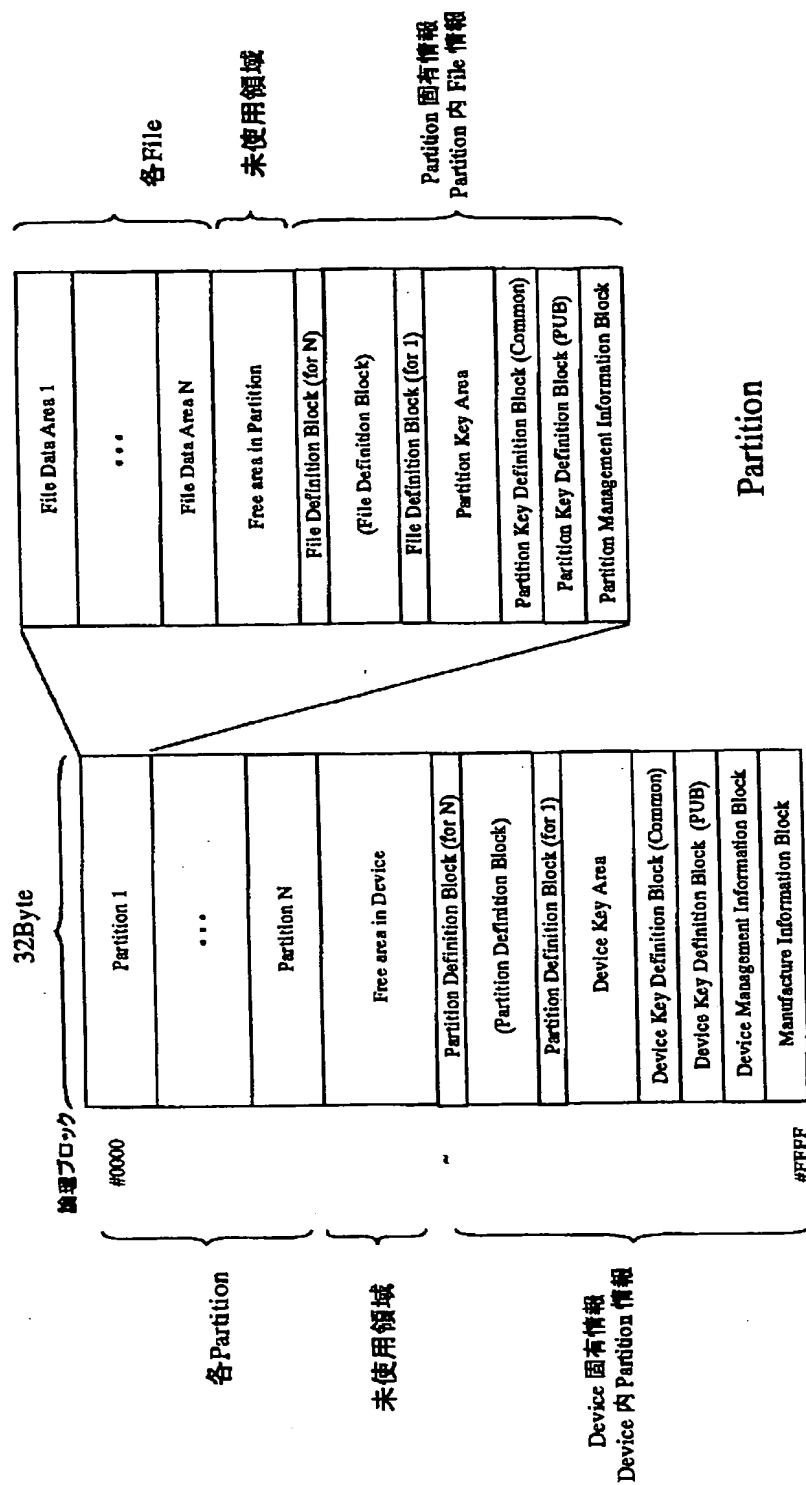


【図11】

## Certificate Format

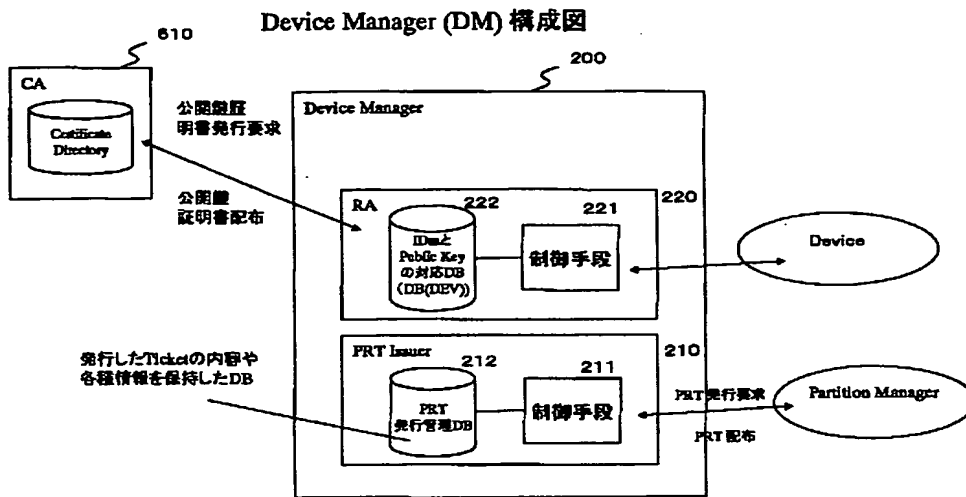
証明書のバージョン番号
発行局(認証局)が割り付ける証明書通し番号(SN)
署名アルゴリズム識別子フィールド: アルゴリズムとパラメータ
発行局(認証局)の名称
証明書の有効期限フィールド: 開始日時、終了日時
公開鍵証明書の利用者名(Subject)
利用者の公開鍵フィールド: 鍵アルゴリズムと鍵情報(鍵そのもの)
オプション領域 (属性など)
発行局(認証局)署名

# Device Memory Format

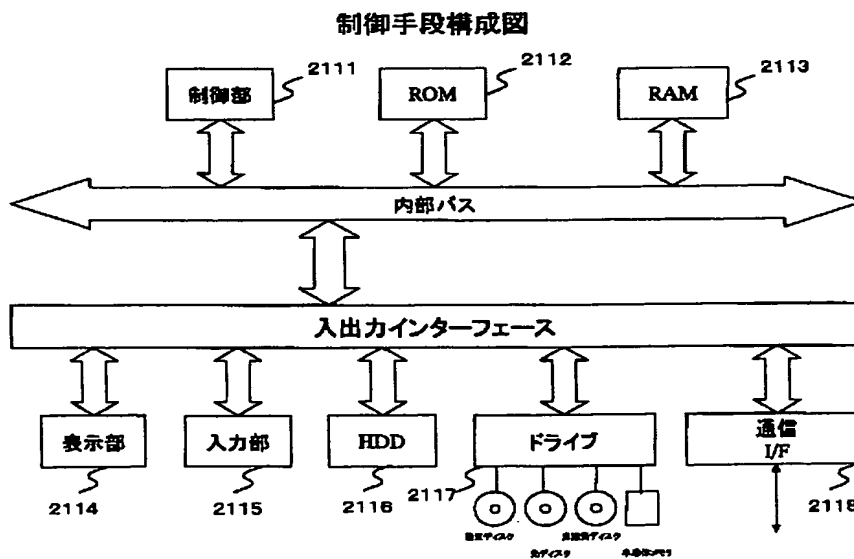


【図 6】

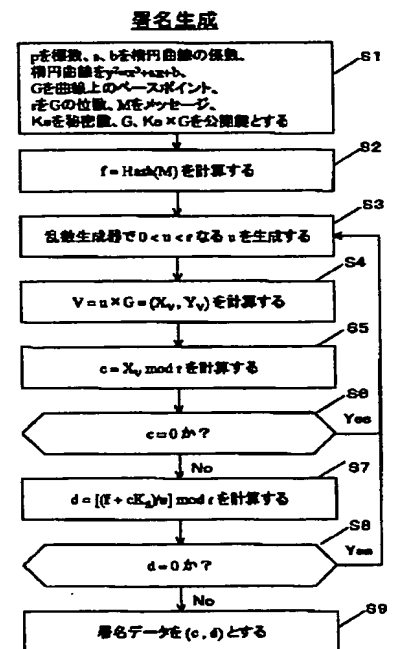
【図7】



【図8】

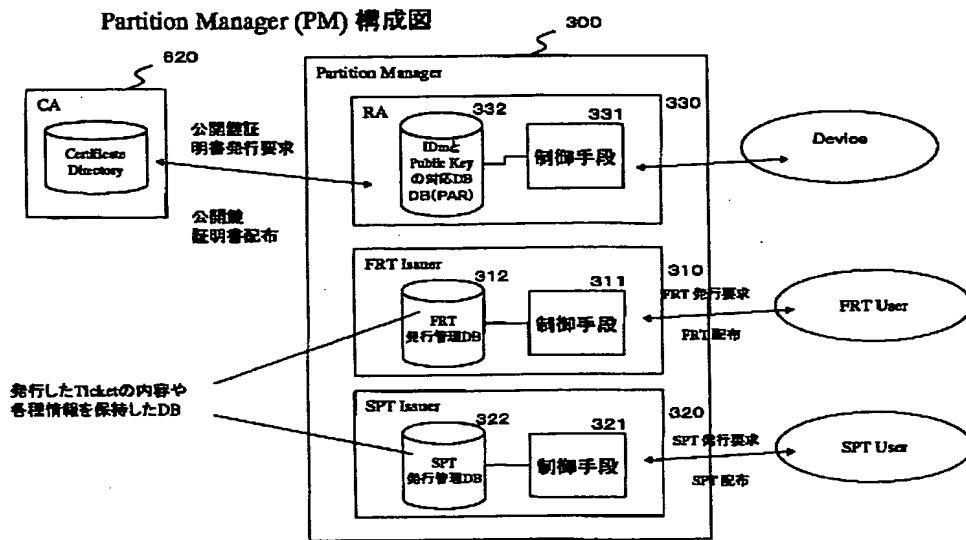


【図12】

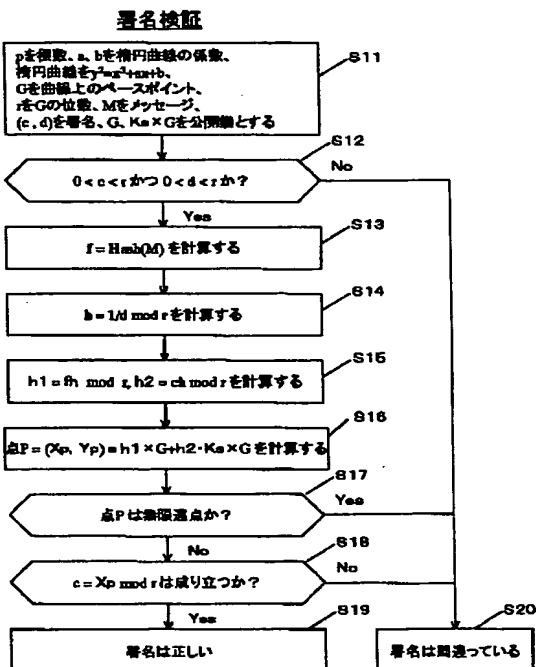


署名生成(IEEE P1363/D13)

【図9】



【図13】



署名検証 (IEEE P1363/D13)

【図18】

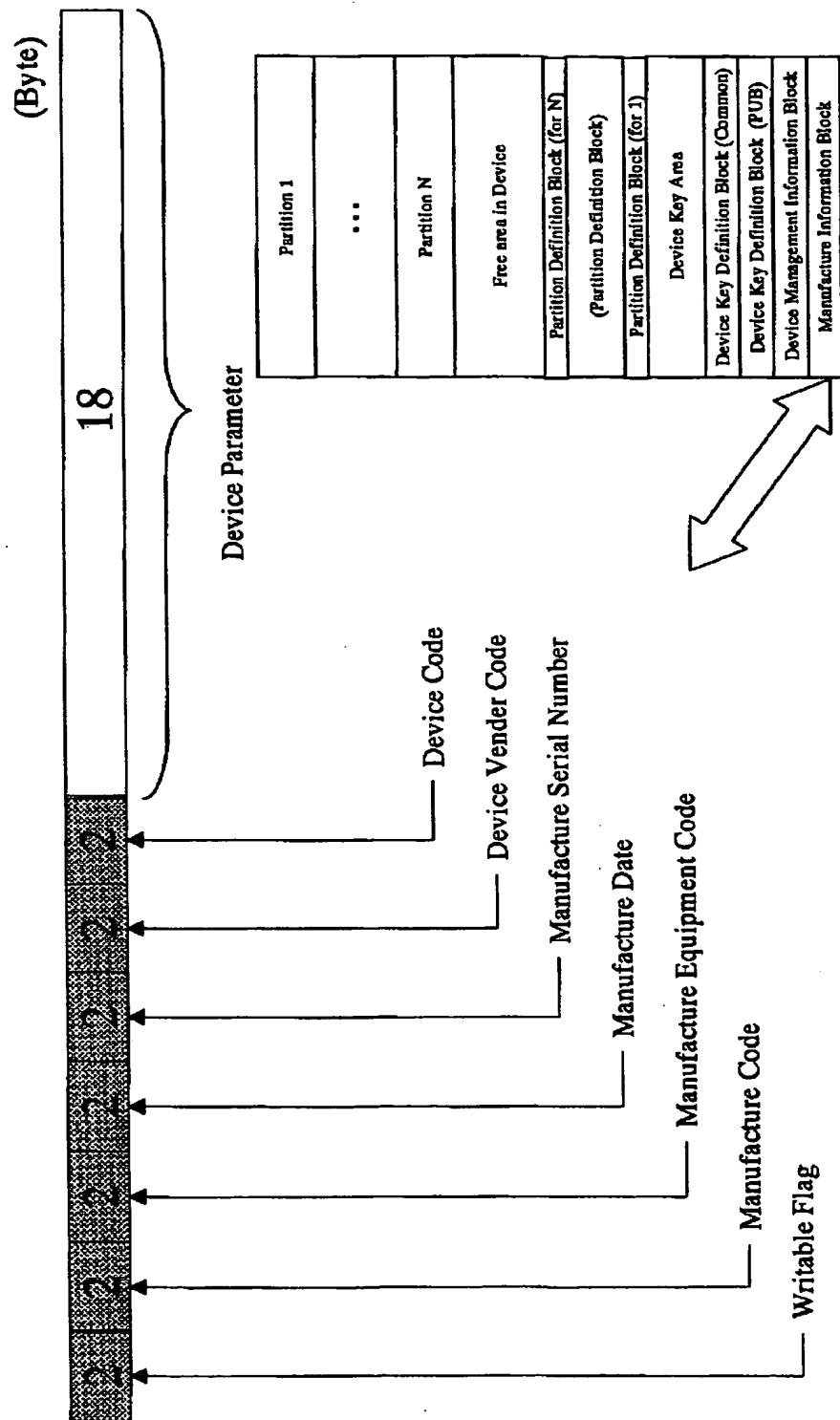
## Device Key Area

Ver	IRL DEV
Ver	CRL DEV
Ver	Kdut DEV4
Ver	Kdut DEV3
Ver	Kdut DEV2
Ver	Kdut DEV1
Ver	Kprt
Ver	CERT DEV
Ver	PRI DEV
Ver	PARAM DEV
Ver	PUB CA(DEV)
Ver	Kauth DEV B
Ver	MKauth DEV A

各項目ごとに、Version 情報を持つ

Device Key Area  
(可変長)

Manufacture Information Block

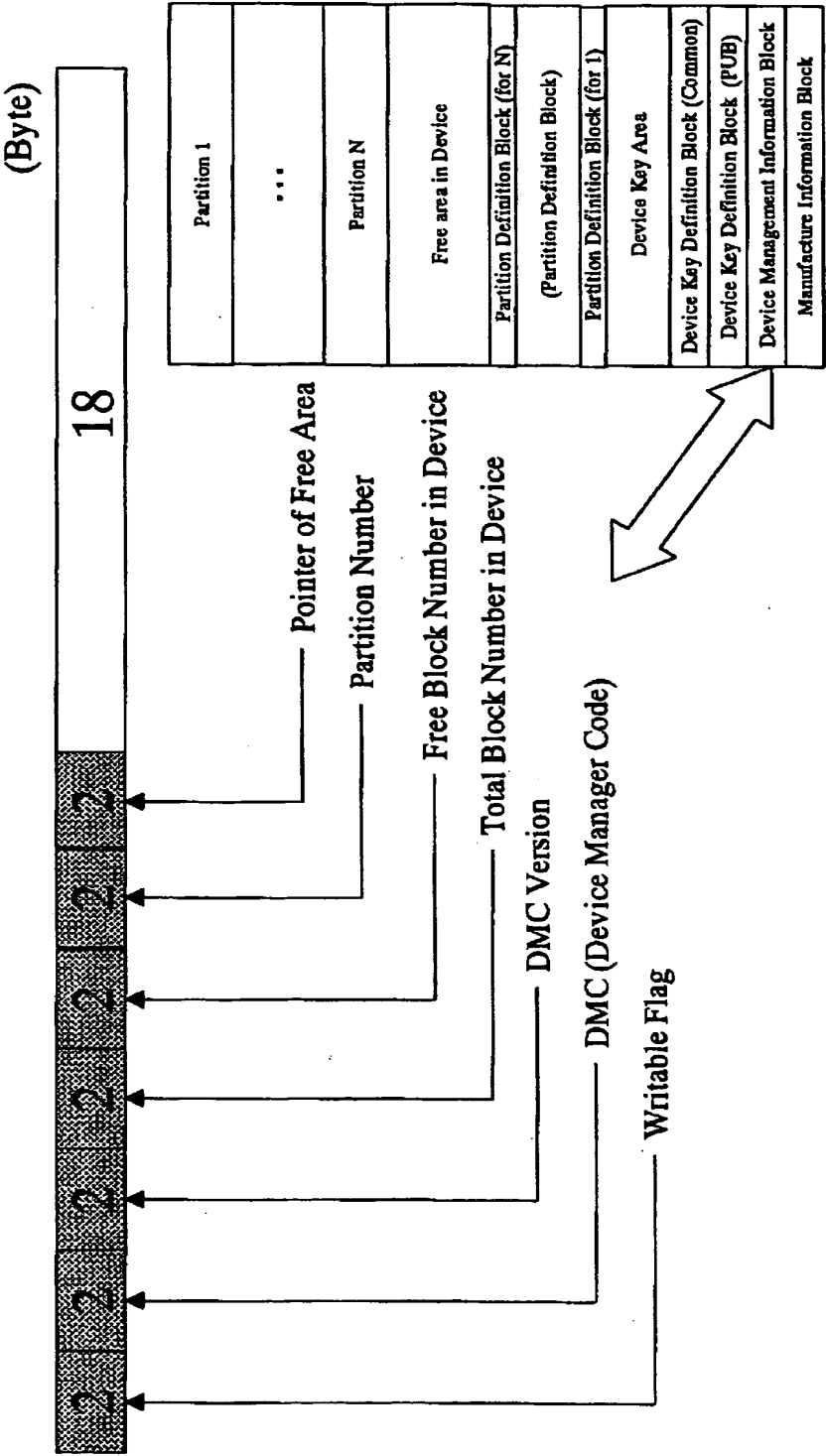


【図 1 4】

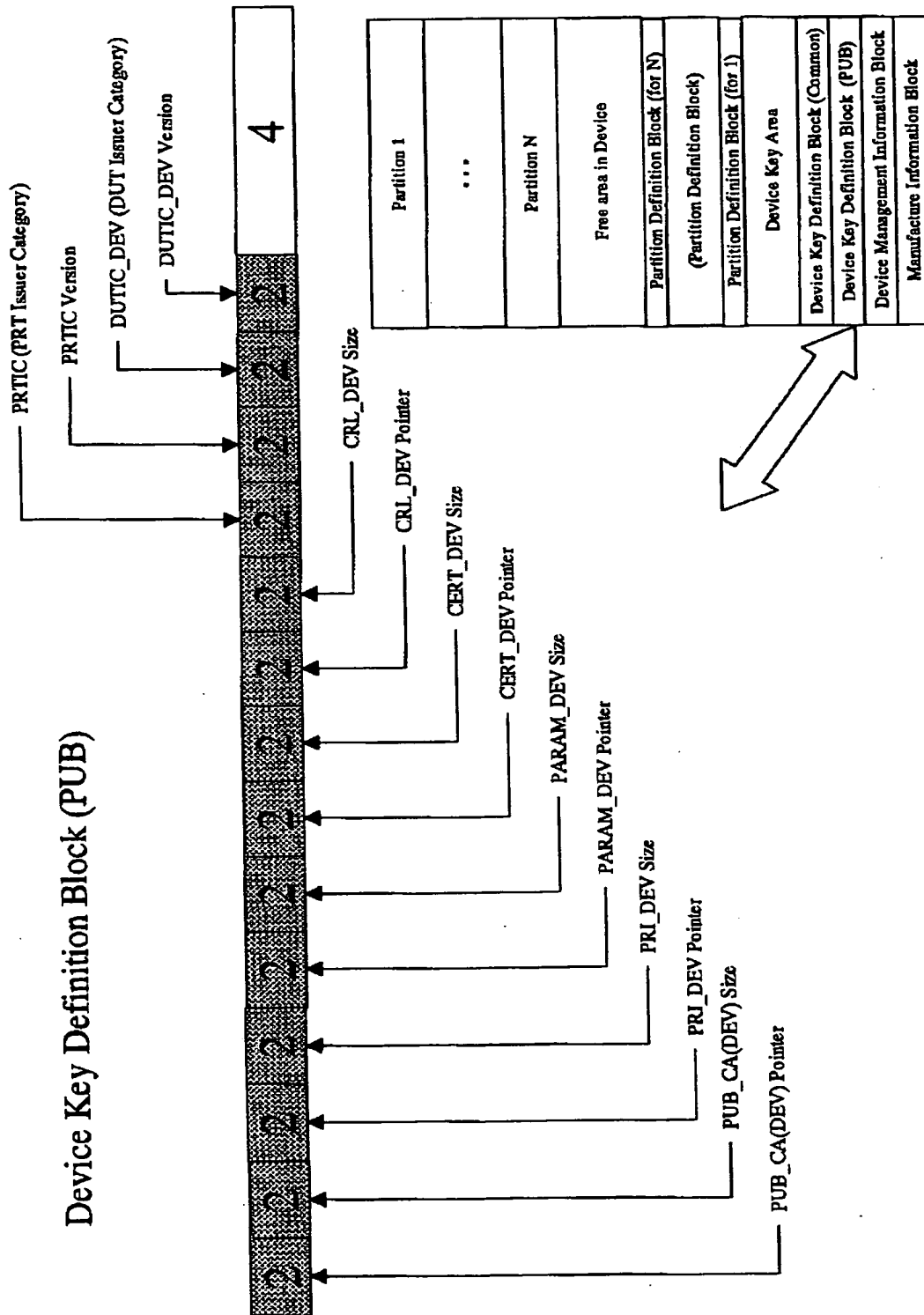


【図15】

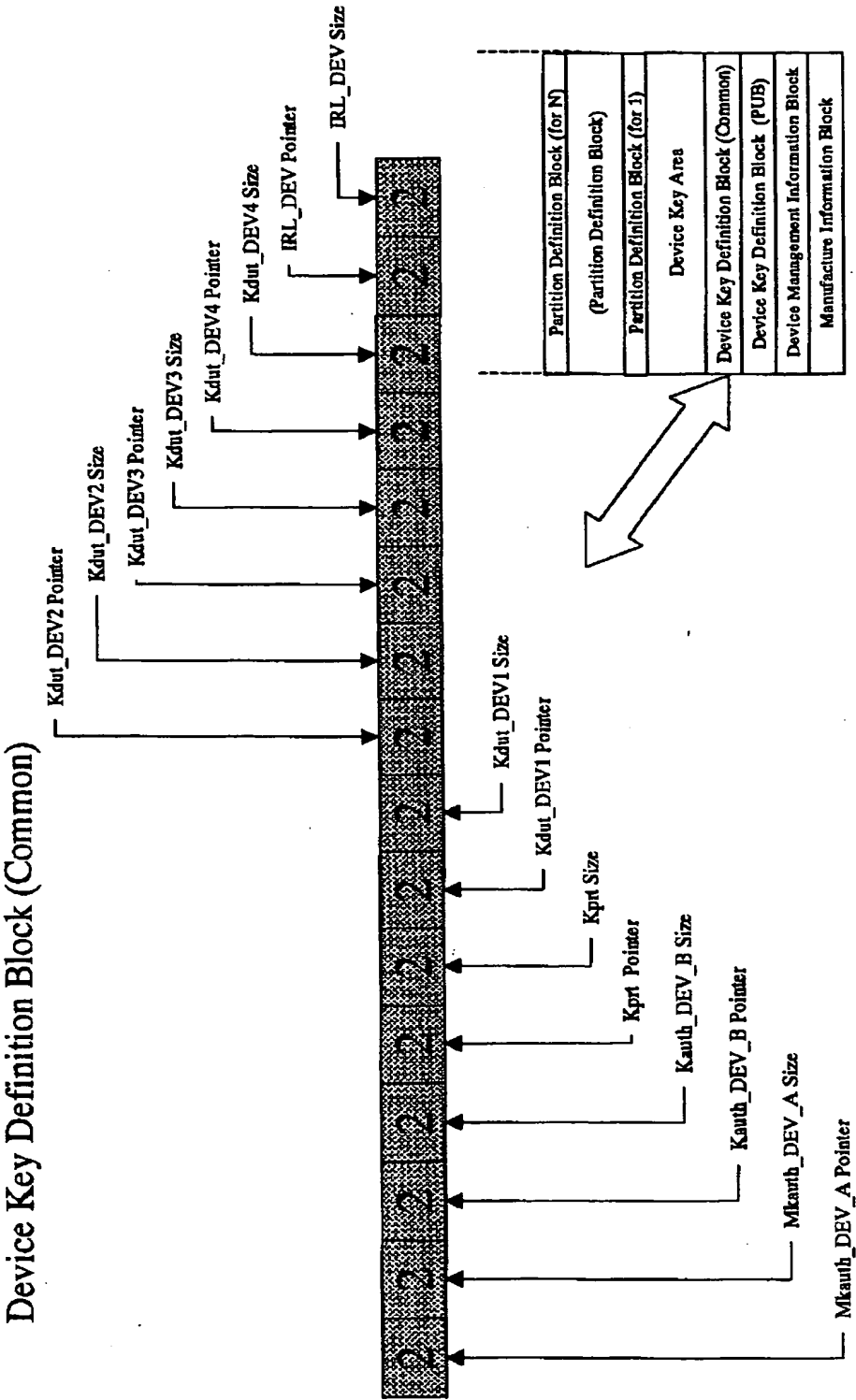
Device Management Information Block



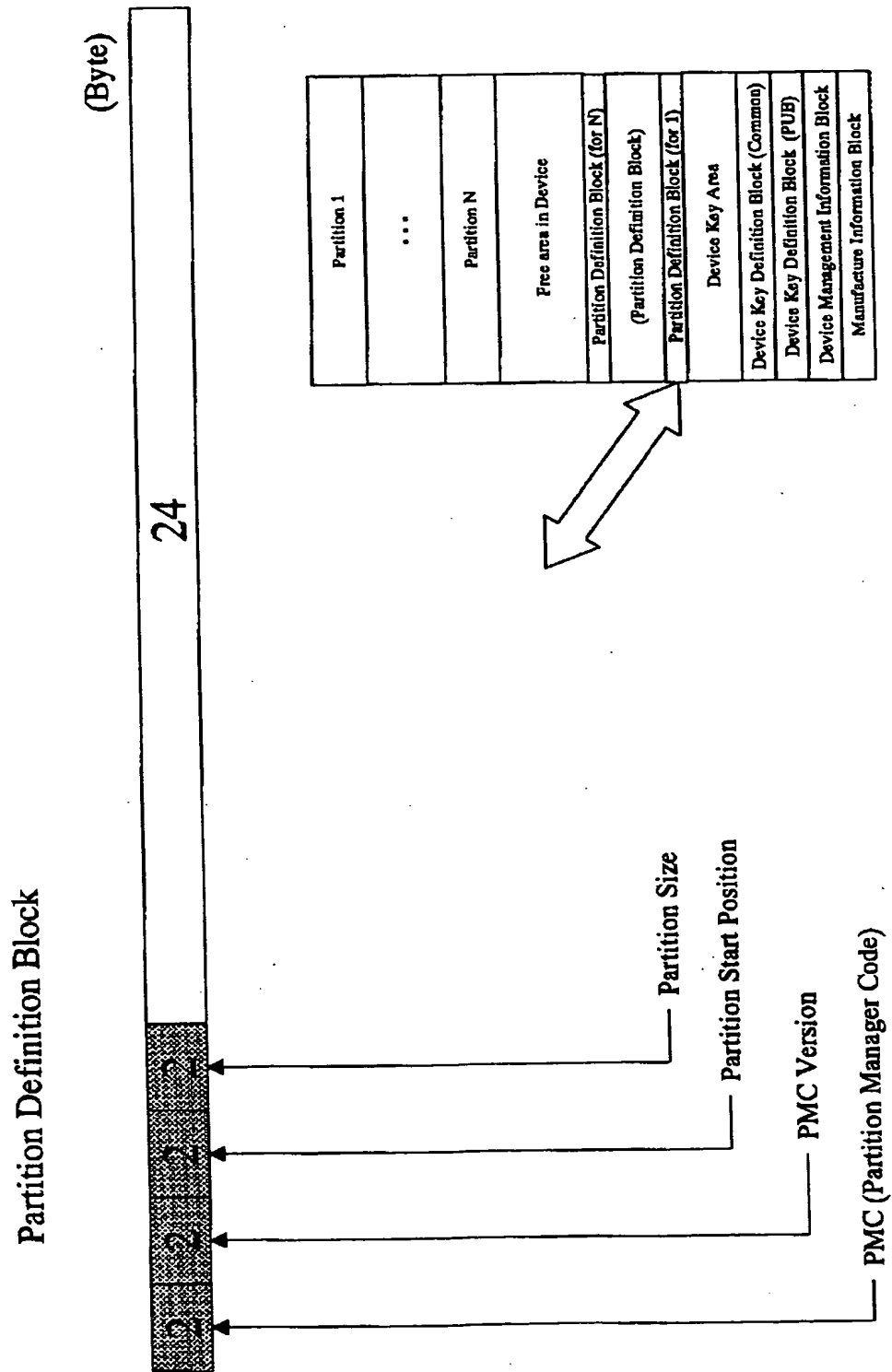
【図 16】



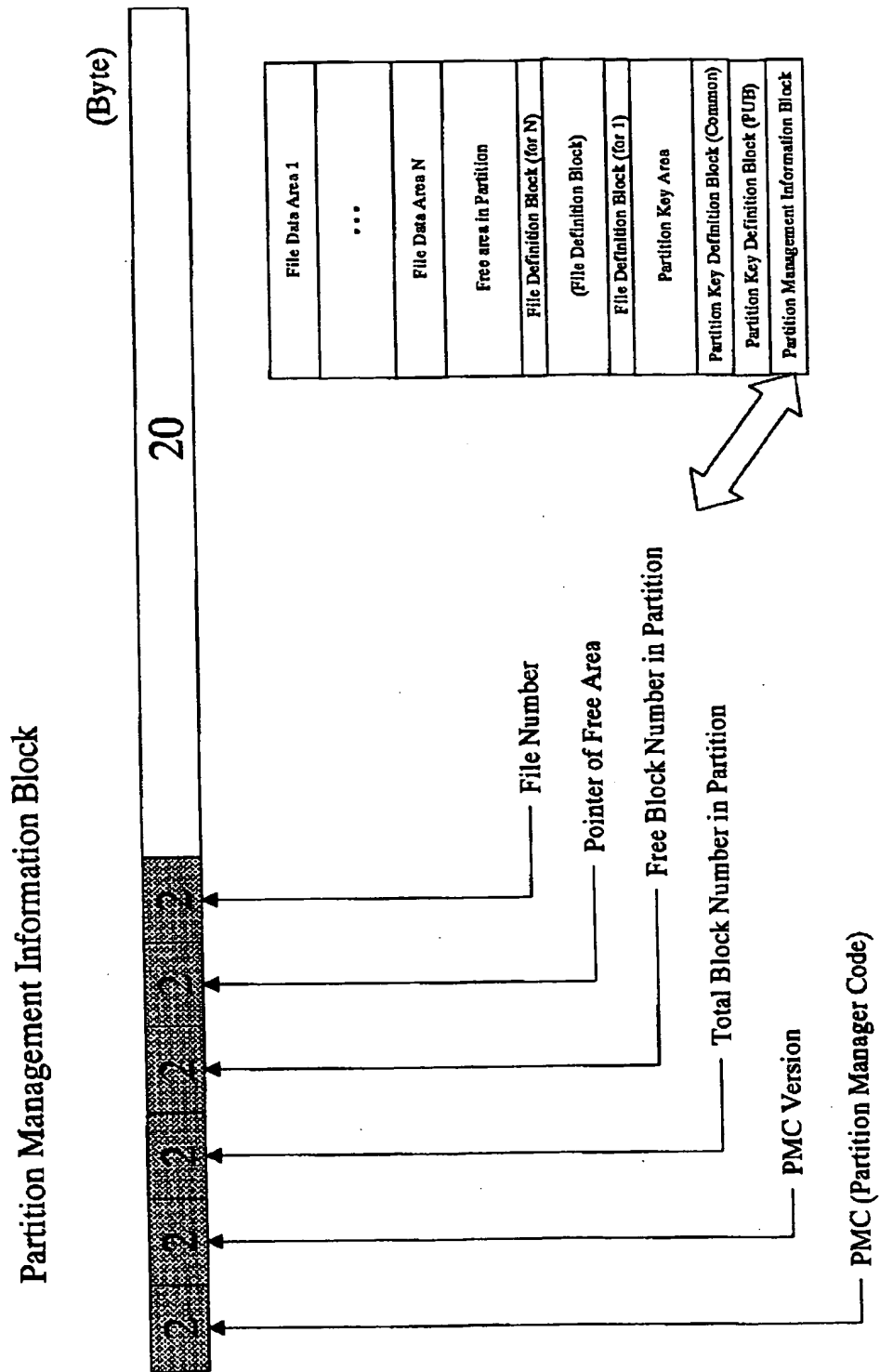
【図 17】



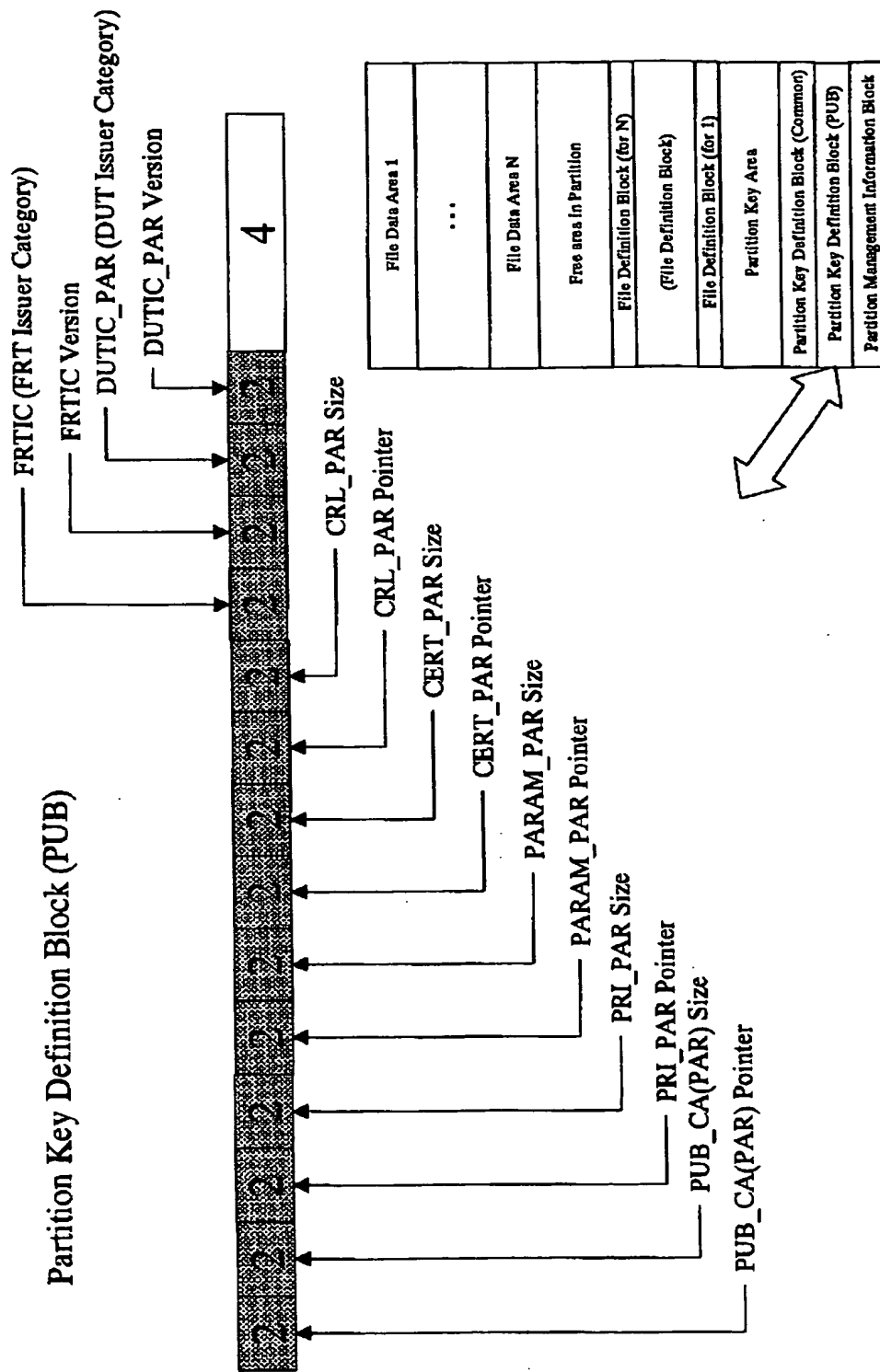
【図 19】



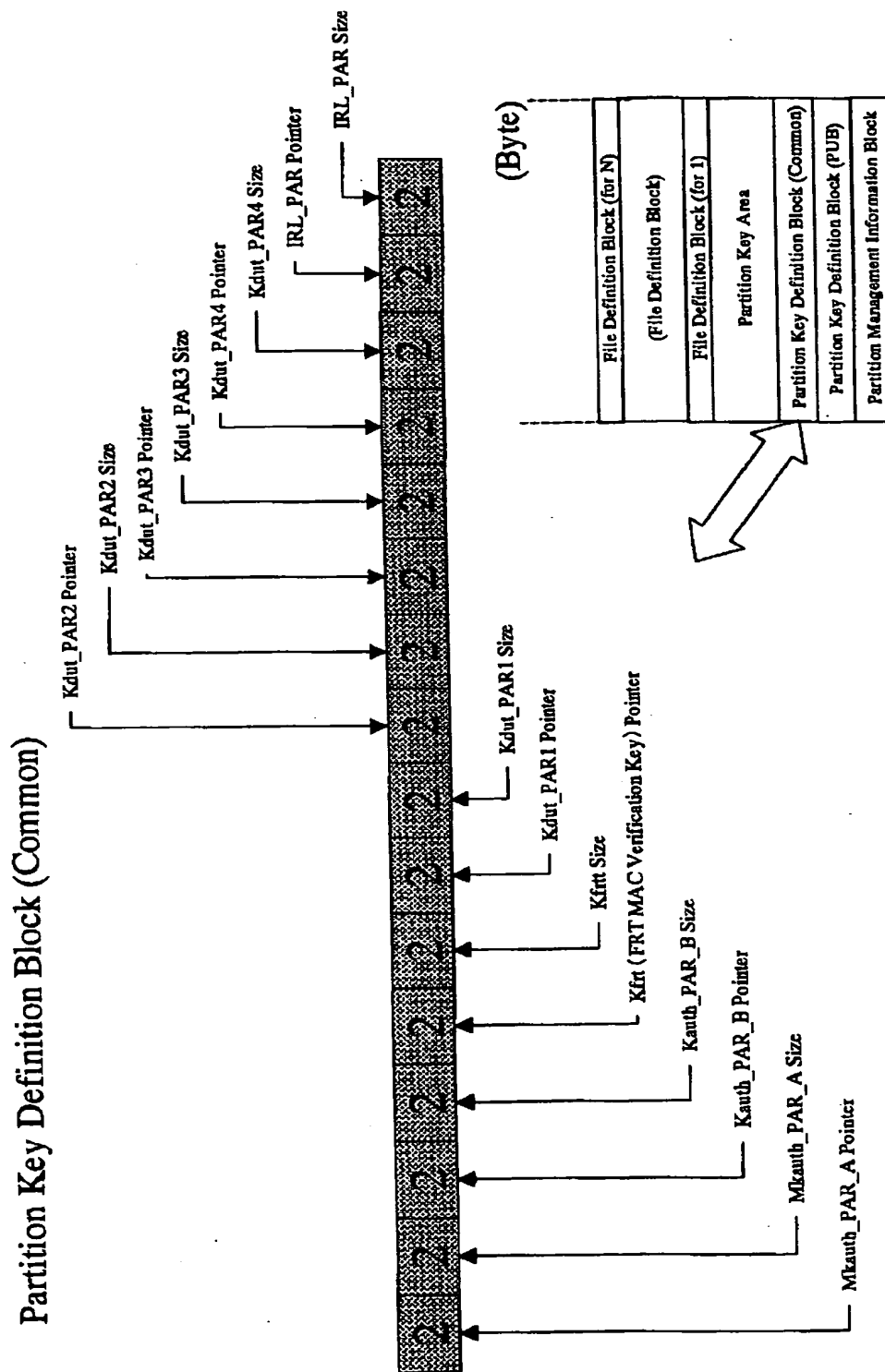
【図 20】



【図 2 1】



【図 2 2】



【図23】

## Partition Key Area

Partition Key Area  
(可変長)

Ver	IRL PAR
Ver	CRL PAR
Ver	Kdut PAR4
Ver	Kdut PAR3
Ver	Kdut PAR2
Ver	Kdut PAR1
Ver	Kft
Ver	CERT PAR
Ver	PRI PAR
Ver	PARAM PAR
Ver	PUB CA(PAR)
Ver	Kauth PAR B
Ver	MKauth PAR A

各項目ごとに、Version 情報を持つ

【図25】

## File Structure

Type Code	Structure Type
0001	Random
0002	Purse
0003	Cyclic
0004	Log
0005	Key
0006	複合ファイル1
0007	複合ファイル2

【図26】

## Partition Registration Ticket (PRT) Format

Ticket Type (=PRT)
Format Version
Ticket Issuer (=DMC)
Serial Number
Size of Ticket
Authentication Flag
Ticket User の所属
Authentication Type
Ticket User の識別子
PMC
PMC Version
Operation Type
Partition Size
Integrity Check Type
Integrity Check Value

【図27】

## File Registration Ticket (FRT) Format

Ticket Type (=FRT)
Format Version
Ticket Issuer (=PMC)
Serial Number
Size of Ticket
Authentication Flag
Ticket User の所属 (Group)
Authentication Type
Ticket User の識別子
SPTIC
SPTIC Ver
File ID
Operation Type
File Size
File Structure Type
Acceptable Authentication Type
Kept Encrypted
Integrity Check Type
Integrity Check Value





【図 28】

## Service Permission Ticket (SPT) Format (1)

Ticket Type (=SPT)
Format Version
Ticket Issuer (=PMC)
Serial Number
Size of Ticket
Authentication Flag
Ticket User の所属 (Group)
Authentication Type
Ticket User の識別子
File ID
File Access Mode
Integrity Check Type
Integrity Check Value

ペア

【図 29】

## File Access Mode

Access Mode Code	Access Mode	Access Mode Code	Access Mode	Access Mode Code	Access Mode
0001	Read	0011	Erase	0021	Certificate 登録
0002	Write	0012	Add	0022	Certificate 検証
0003	暗号化 Read	0013	Sub	0023	鍵ペア生成
0004	暗号化 Write	0014	Compare	0024	鍵ペア検証
0005	MAC付 Read	0015	暗号化	0025	相互認証
0006	MAC付 Write	0016	復号	0026	入金系
0007	MAC付 暗号化 Read	0017	署名生成	0027	出金系
0008	MAC付 暗号化 Write	0018	署名検証	0028	
0009	Write Only	0019	MAC生成	0029	
0010	Write Once	0020	MAC検証	0030	

【図 51】

## Device 内認証テーブル

Group	公開鍵方式認証者情報	Session Key	共通鍵方式認証者情報	Session Key
DMC	DN, Serial Number, Category	Kses1	—	—
PMC1	—	—	ID_RW	Kses2
PMC2	DN, Serial Number, Category	Kses3	ID_RW	Kses4

【図30】

## File Structure, Access Mode &amp; Command

File Structure	Access Mode	Acceptable Command	File Structure	Access Mode	Acceptable Command
Random	Read	Read	複合ファイル (電子マネー)	入金系	Deposit
	Write	Write		出金系	Withdraw Make Receipt Read Receipt
	暗号化 Read	EncRead			
	暗号化 Write	EncWrite			
	MAC付 Read	MacRead			
	MAC付 Write	MacWrite			
	MAC付 暗号化 Read	EncMacRead			
	MAC付 暗号化 Write	EncMacWrite			

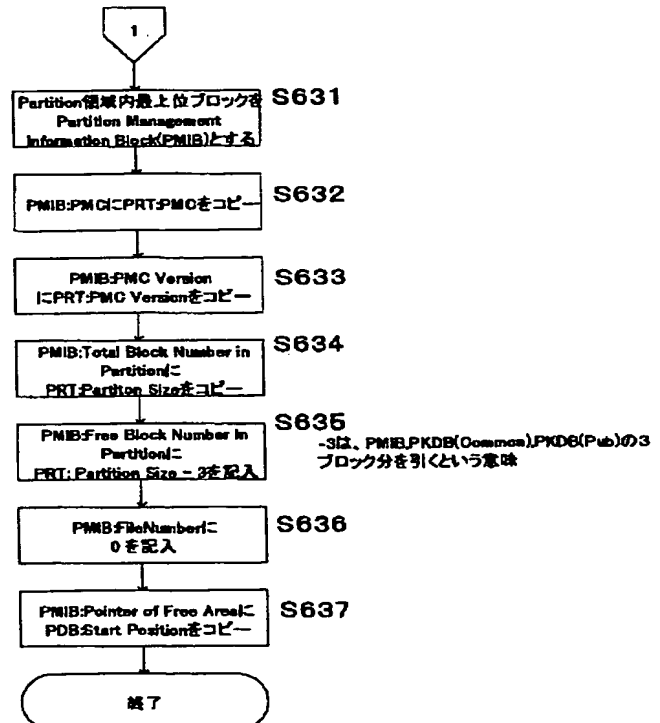
【図31】

## Service Permission Ticket (SPT) Format (2)

Ticket Type (=SPT)
Format Version
Ticket Issuer (=PMC)
Serial Number
Size of Ticket
Authentication Flag
Ticket User の所属 (Group)
Authentication Type
Ticket User の識別子
File ID
File Access Mode
Group of Target File
Target File ID
Read/Write Permission
Integrity Check Type
Integrity Check Value

1-セット

【図61】



# Data Update Ticket (DUT) Format

Ticket Type (=DUT(DEV))
Format Version
Ticket Issuer (=DMC)
Serial Number
Size of Ticket
Ticket User の所属
Ticket User の識別子
Authentication Type
Encrypted Flag
Old Data Code
Data Version Rule
Data Version Condition
Size of New Data
New Data
New Data Version
Integrity Check Type
Integrity Check Value

DUT(DEV)

【図 3 2】

Ticket Type (=DUT(PAR))
Format Version
Ticket Issuer (=PMC)
Serial Number
Size of Ticket
Ticket User の所属
Ticket User の識別子
Authentication Type
Encrypted Flag
Old Data Code
Data Version Rule
Data Version Condition
Size of New Data
New Data
New Data Version
Integrity Check Type
Integrity Check Value

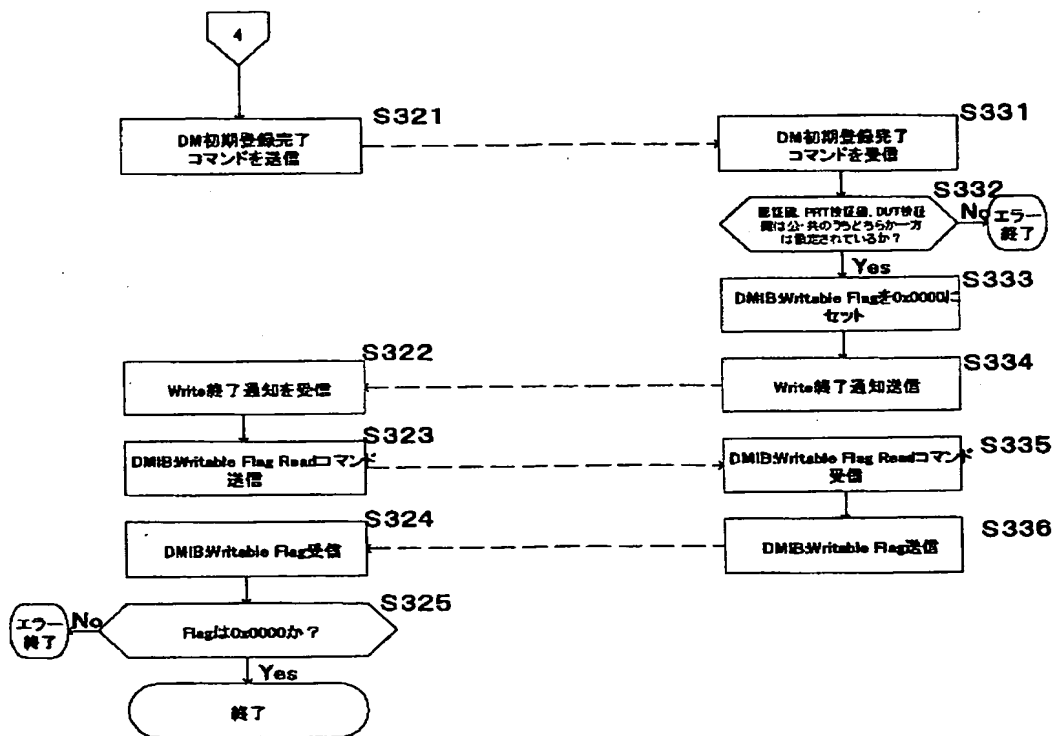
DUT(PAR)

【図33】

## New Data Update Code

Code	New Data	Code	New Data	Code	New Data
0001	DMC	0011	DUTIC_PAR	0021	MKauth_DEV_A
0002	DMC Version	0012	DUTIC_PAR Version	0022	Kauth_DEV_B
0003	PMC	0013	Kdnt_DEV1,2	0023	IRL_DEV
0004	PMC Version	0014	Kdnt_PAR1,2	0024	IRL_PAR
0005	PRTIC	0015	Kprt	0025	
0006	PRTIC Version	0016	Kprt Version	0026	
0007	FRTIC	0017	Kfirt	0027	
0008	FRTIC Version	0018	Kfirt Version	0028	
0009	DUTIC_DEV	0019	Kept	0029	
0010	DUTIC_DEV Version	0020	Kept Version	0030	

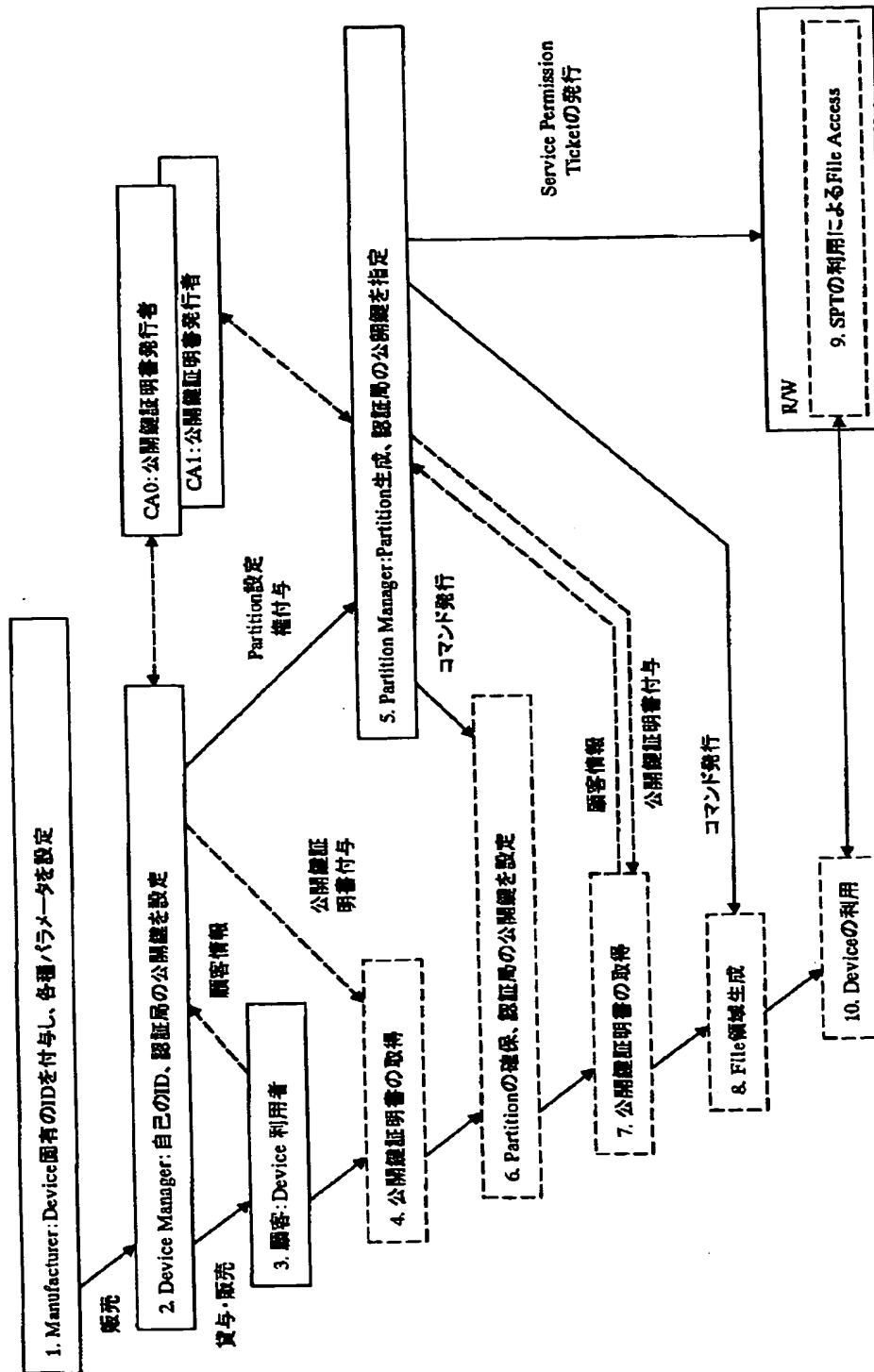
【図40】



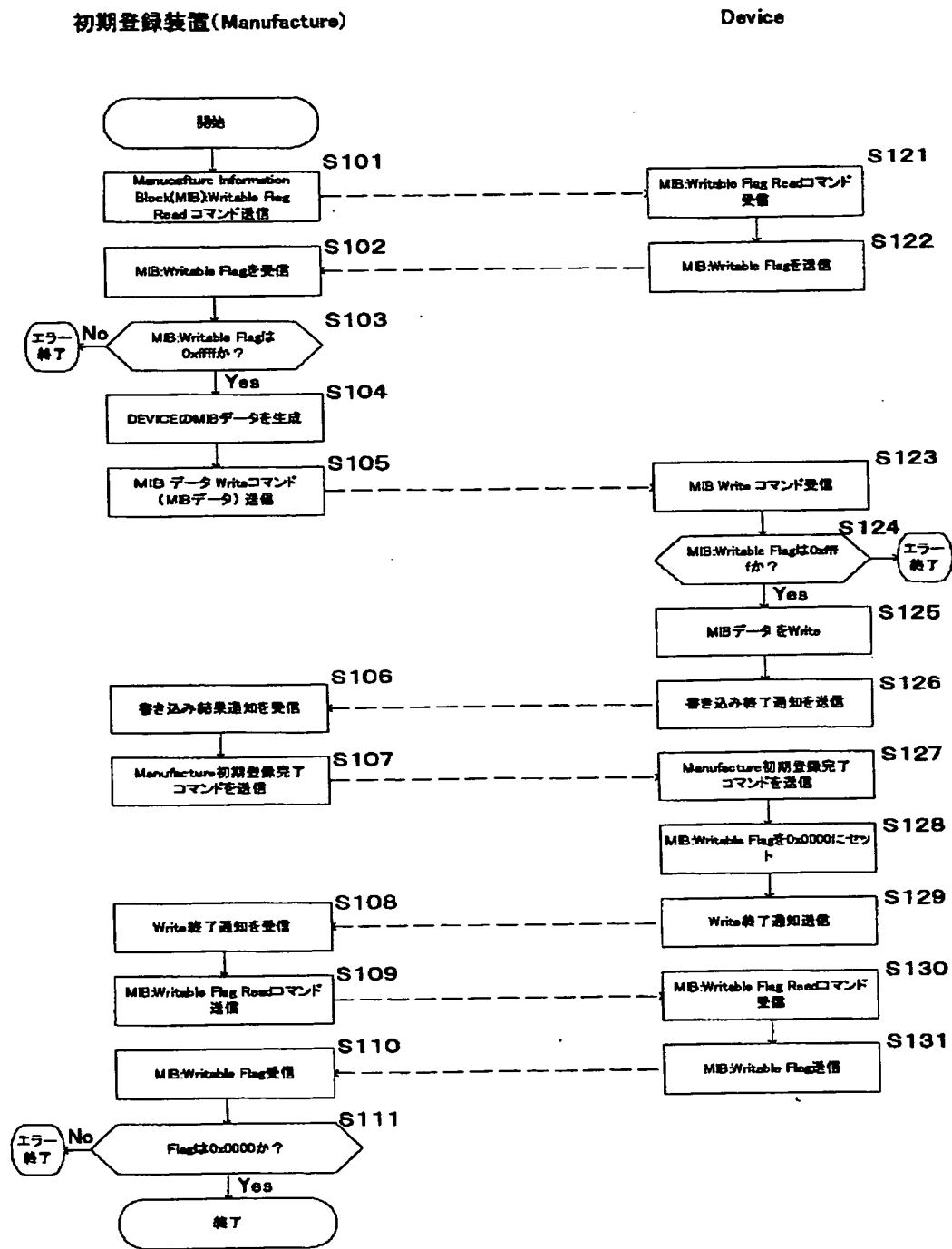
Device初期登録(DM)(続き)

【図34】

## Device 利用までの流れ

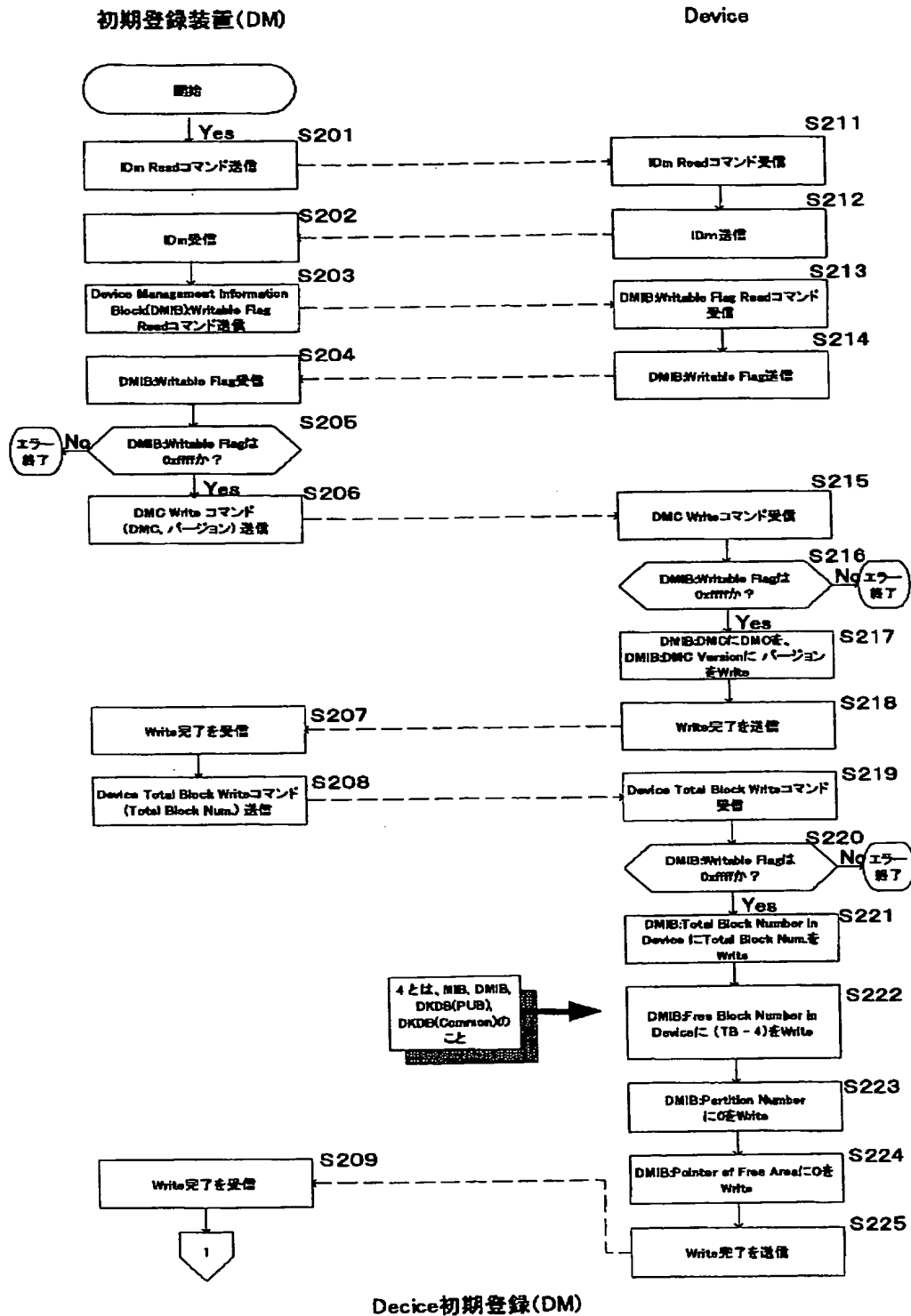


【図35】



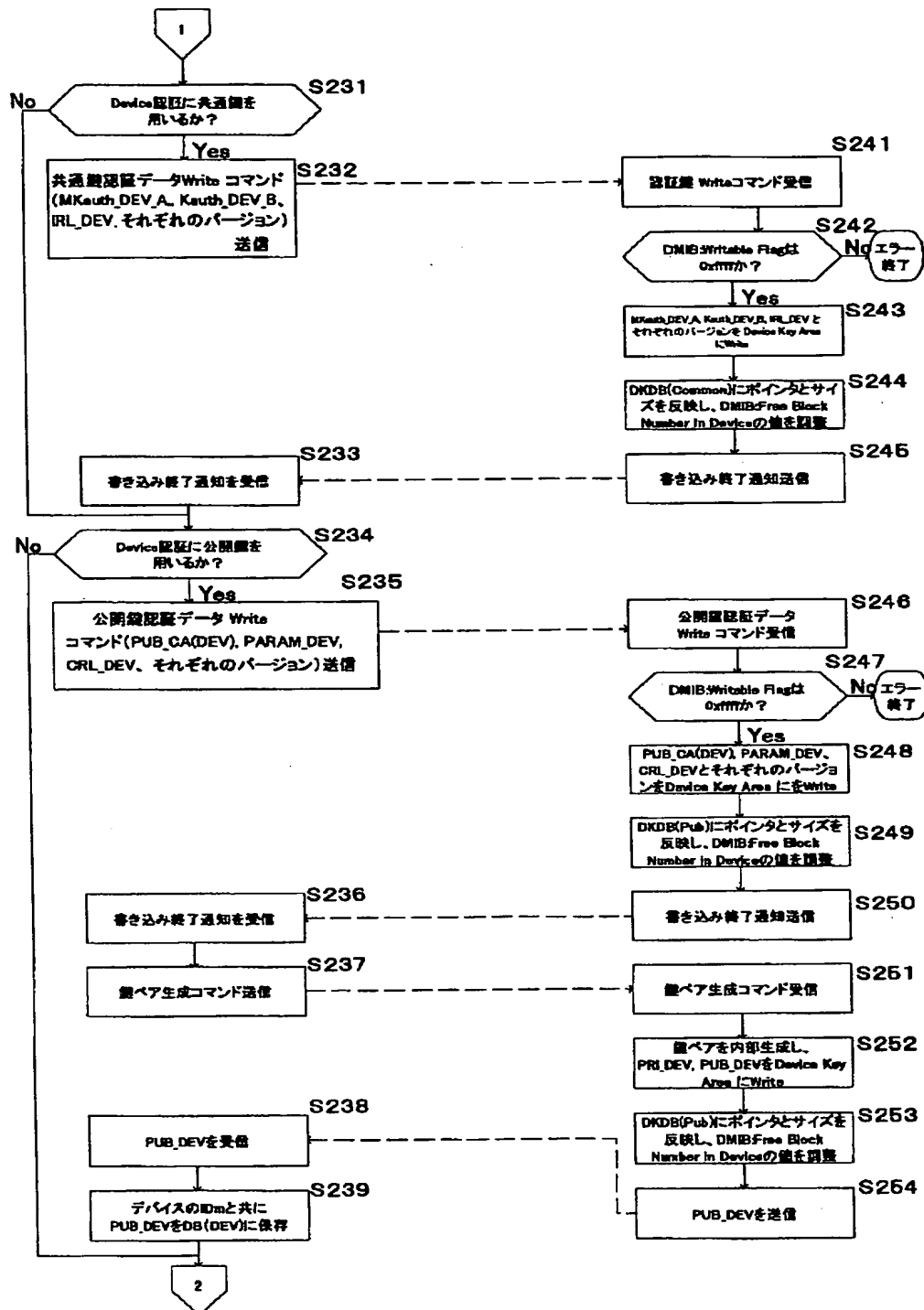
Device初期登録 (Manufacture)

【図36】



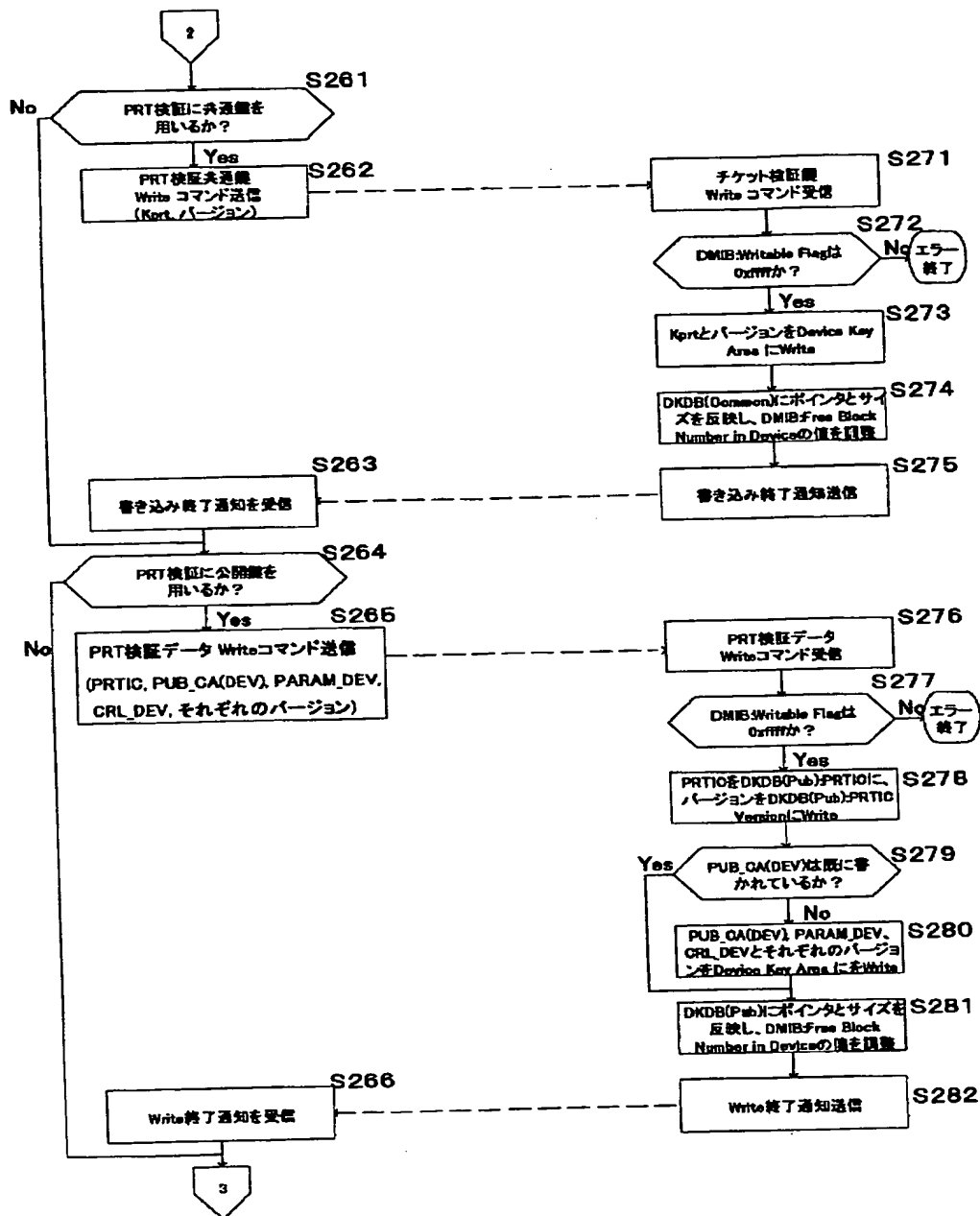


【図37】



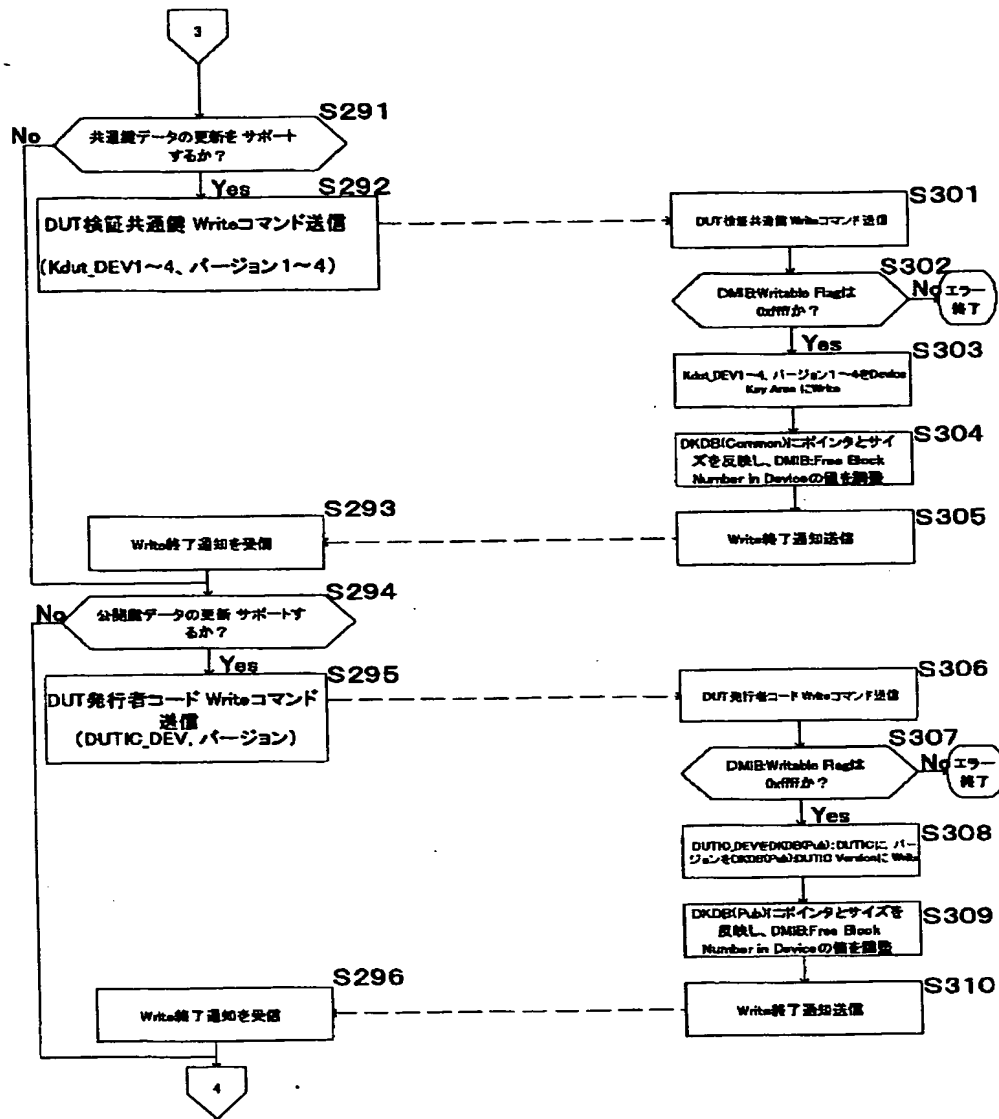
Device 初期登録(DM)(続き)

【図38】



Device初期登録(DM)(続き)

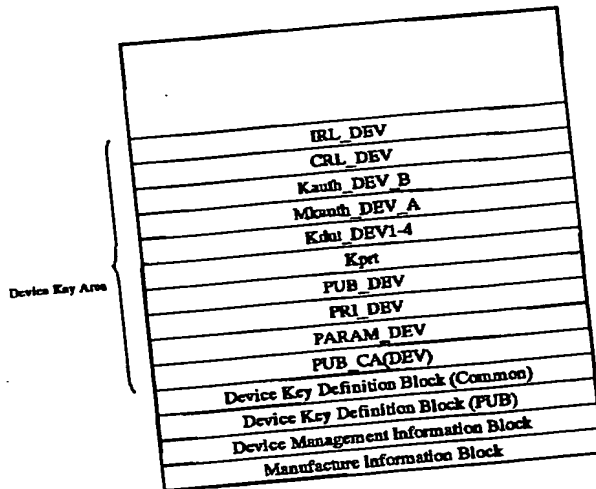
【図39】



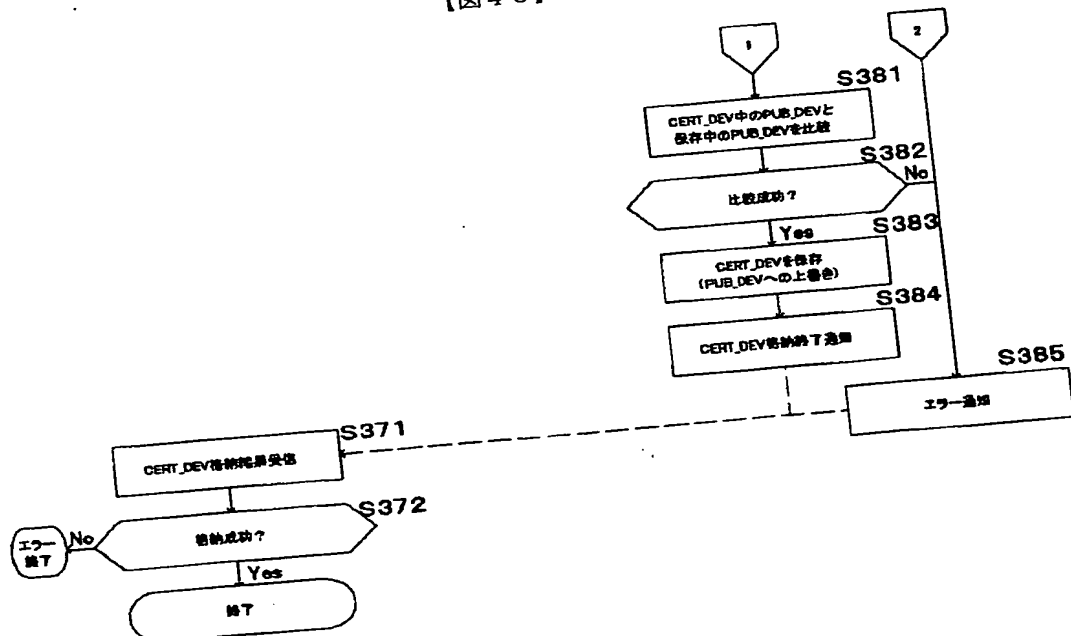
Device初期登録(DM)(続き)

【図41】

初期于一夕書き込み

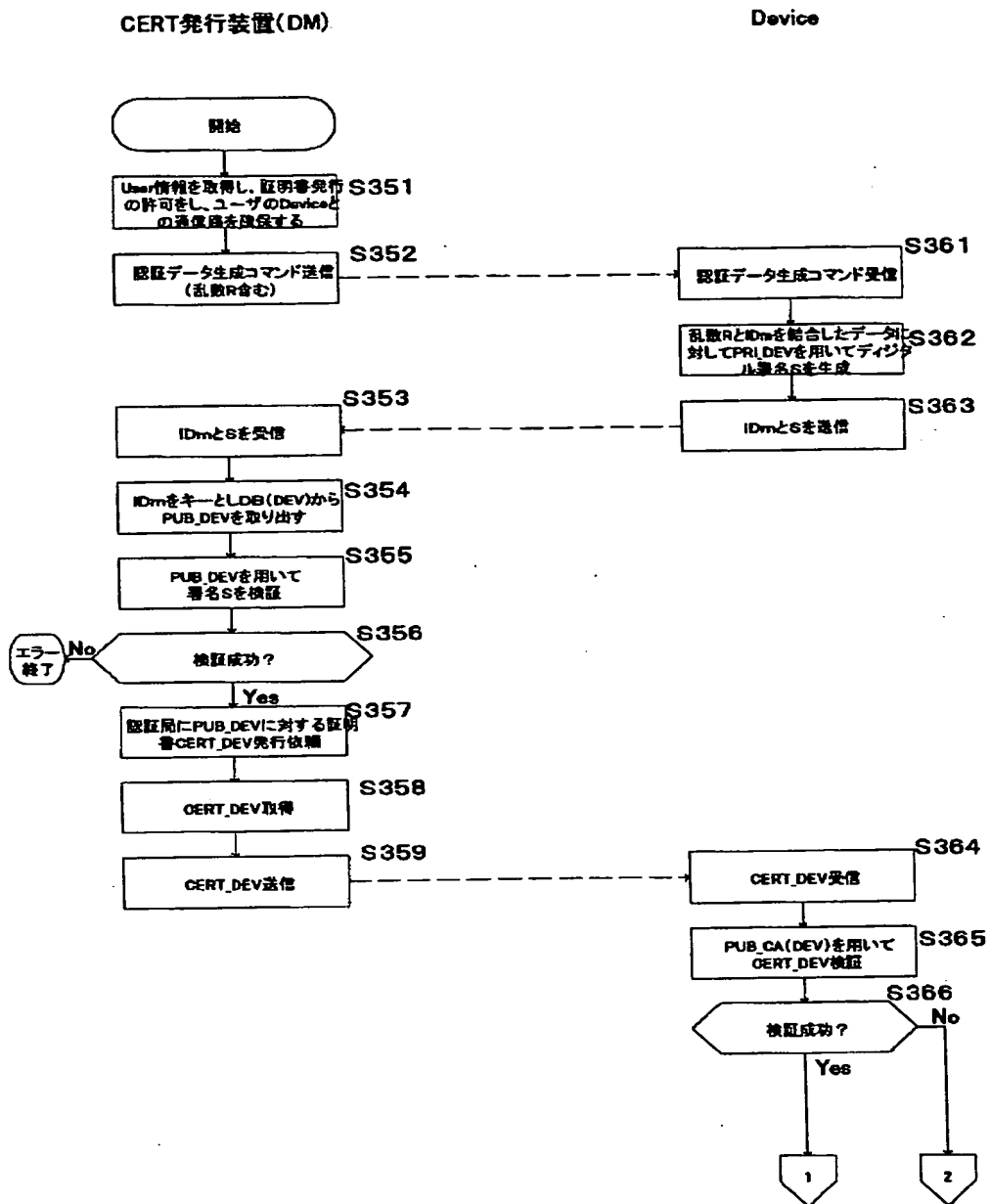


【圖 4 3】



CERT発行(DM)(続き)

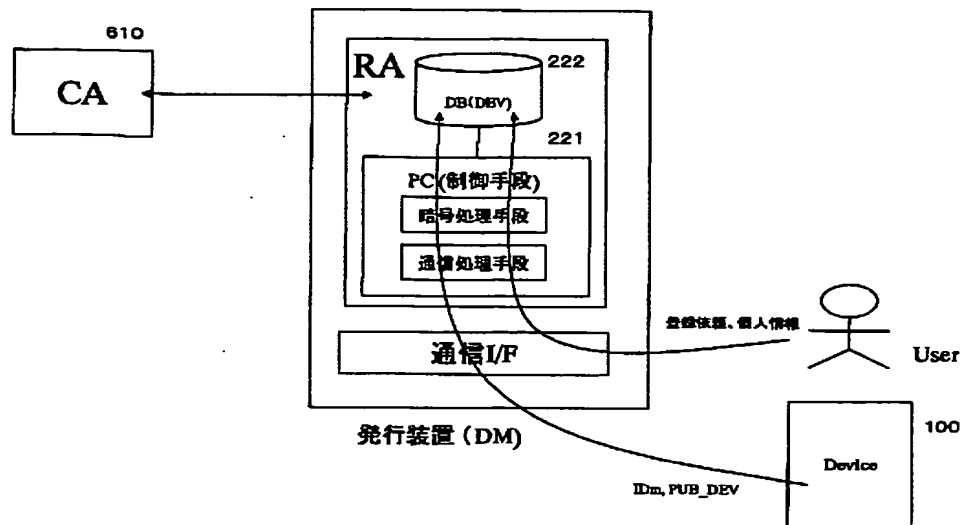
【図42】



CERT発行(DM)

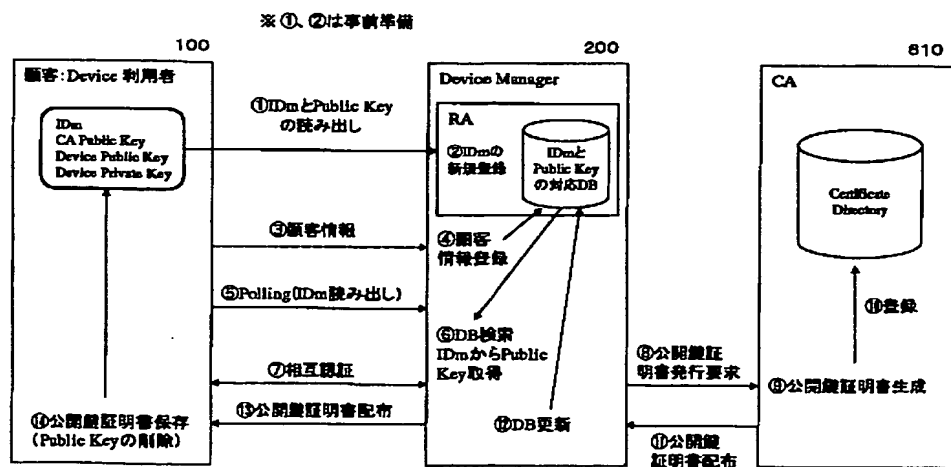
【図44】

## 発行装置 (DM) (=RA) 構成図



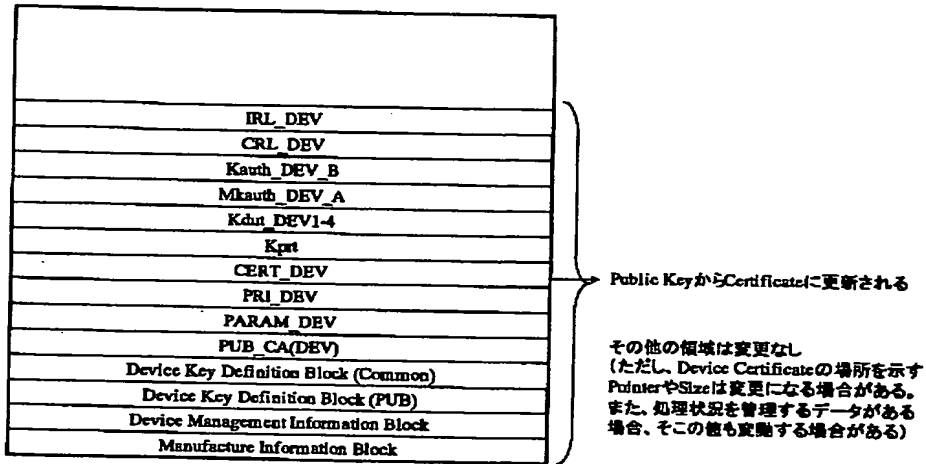
【図45】

## Certificate発行手順



【図46】

## Certificate発行後の状態



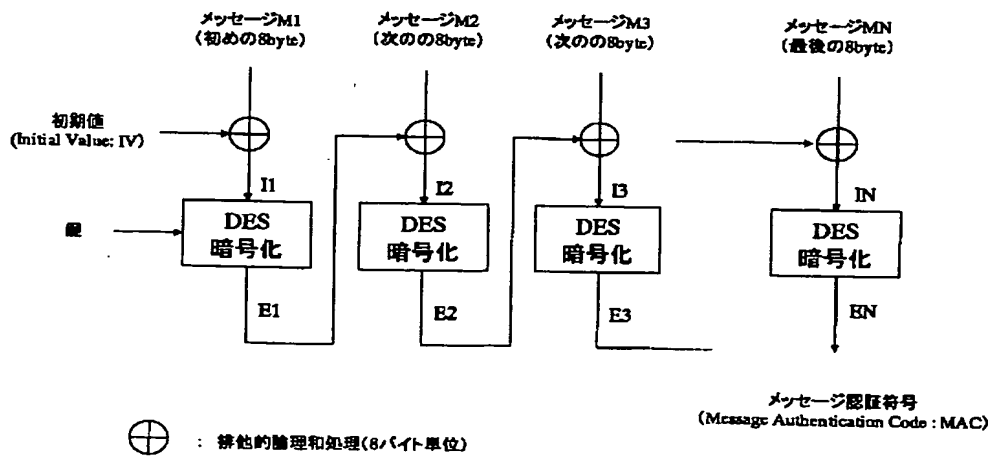
【図52】

## R/W 内認証テーブル

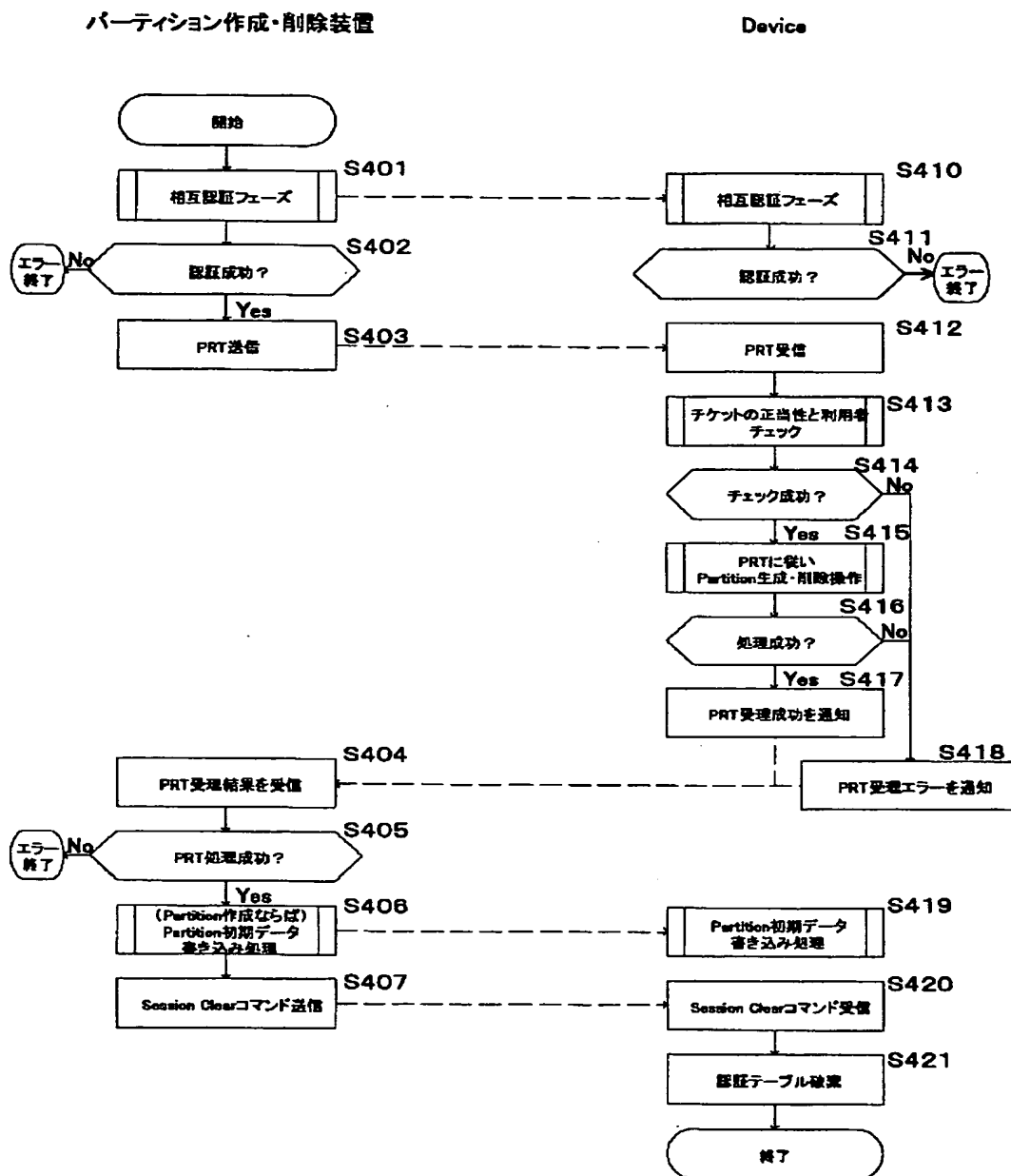
Group	公開鍵方式認証者情報	共通鍵方式認証者情報	Session Key
DMC	DN, Serial Number, Category	—	Kses1
PMC1	—	IDm	Kses2
PMC2	DN, Serial Number, Category	IDm	Kses3

【図59】

## MAC生成方式



【図47】



Partition生成・消去



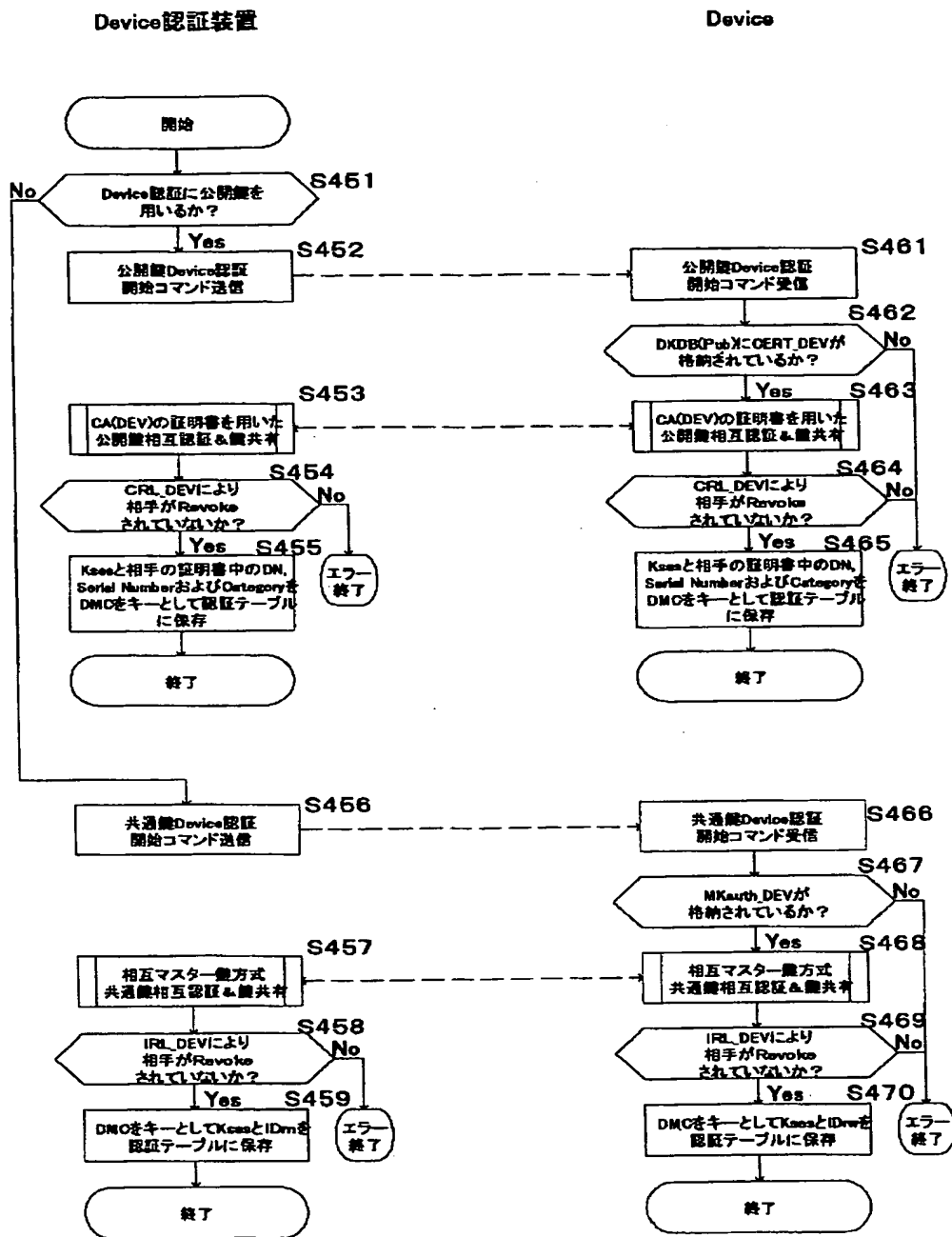
## 認証装置



## File Open テーブル(2)

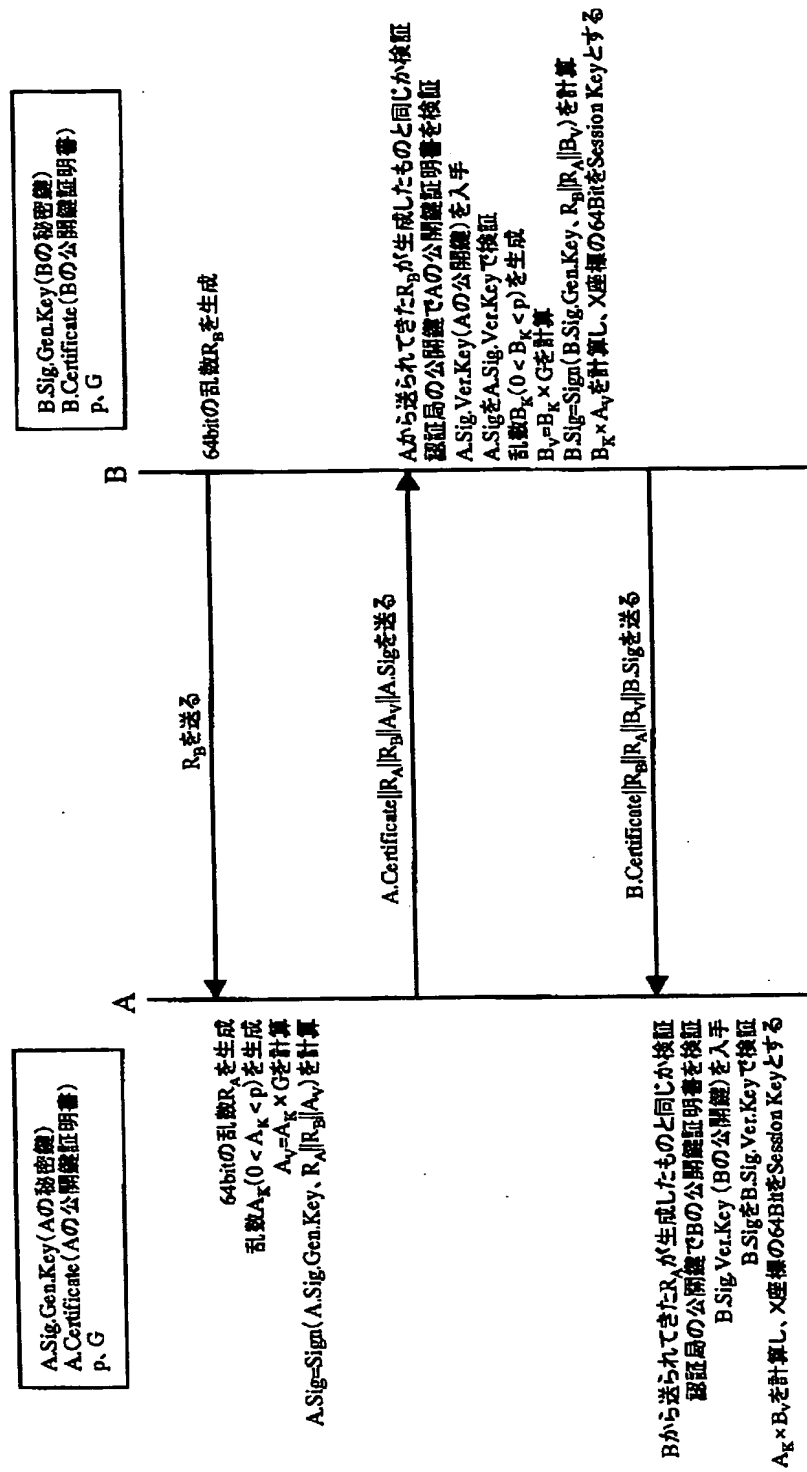
Group	File ID	File Access Mode	Group of Target File	Target File ID	Read/Write Permission
FMC1	0x0001	略号化、複号			
FMC2	0x0002		FMC1	0x0001	Read

【図49】



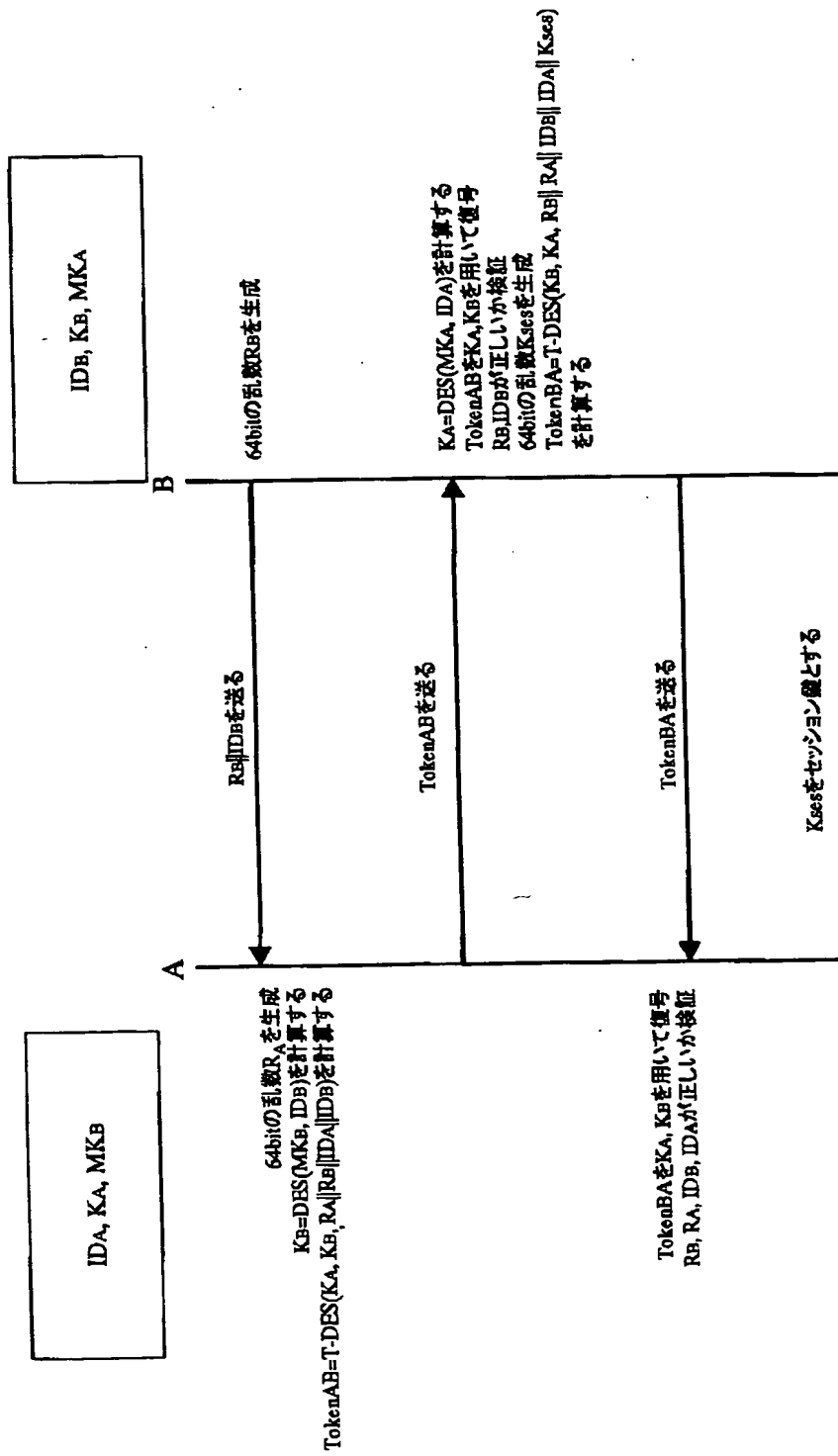
【図50】

## ISO/IEC 9798-3 非対称鍵暗号技術を用いた相互認証および鍵共有方式



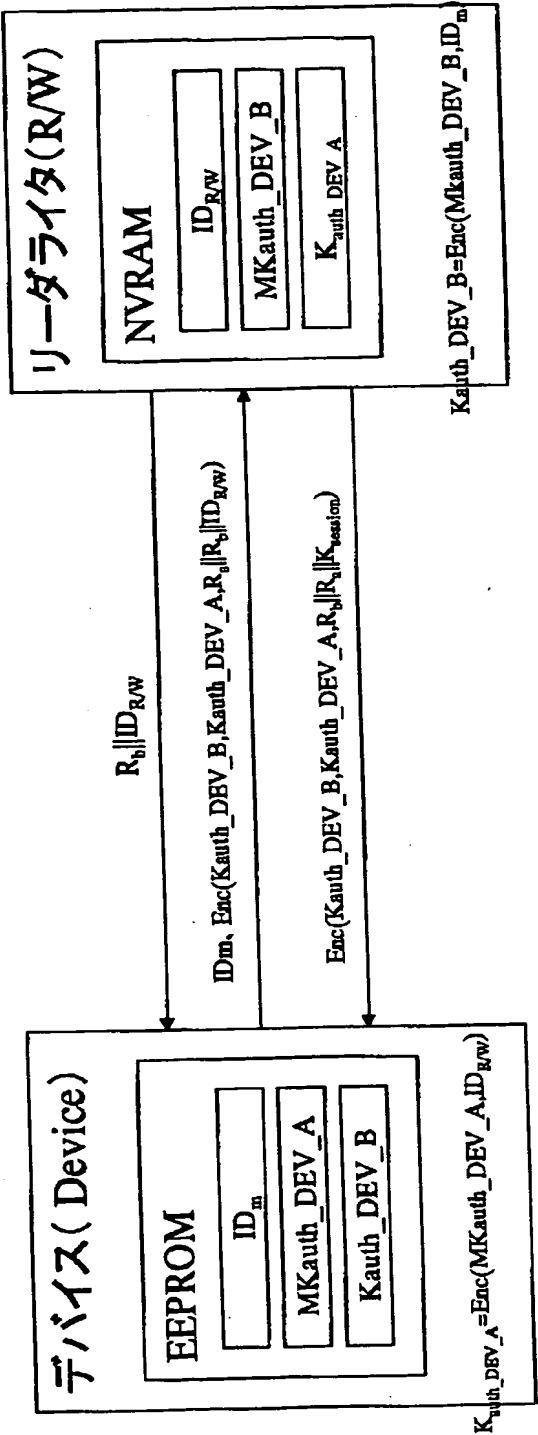
【図53】

## 相互にマスター鍵を用いた鍵共有方式

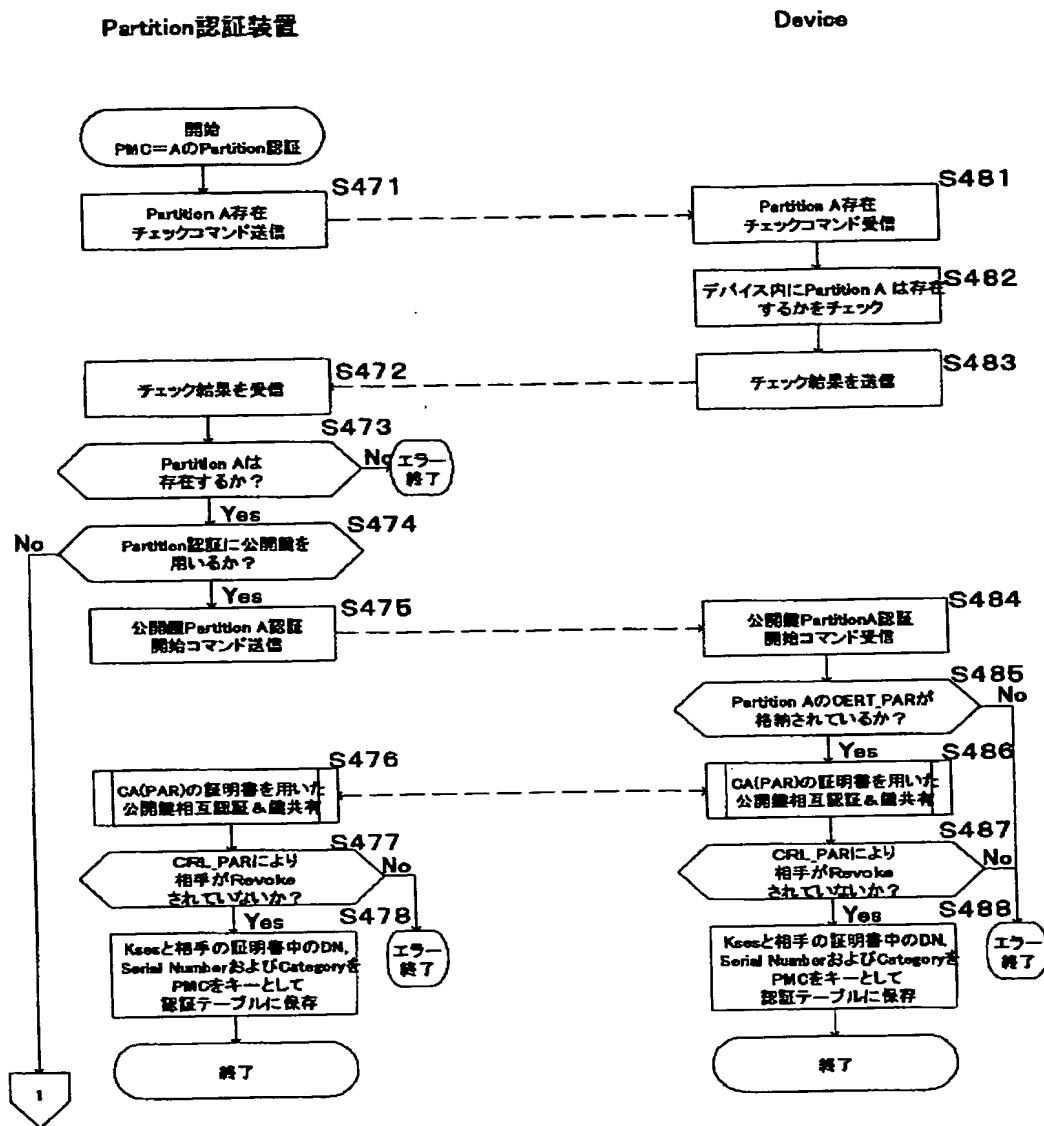


【図54】

双方向個別鍵認証

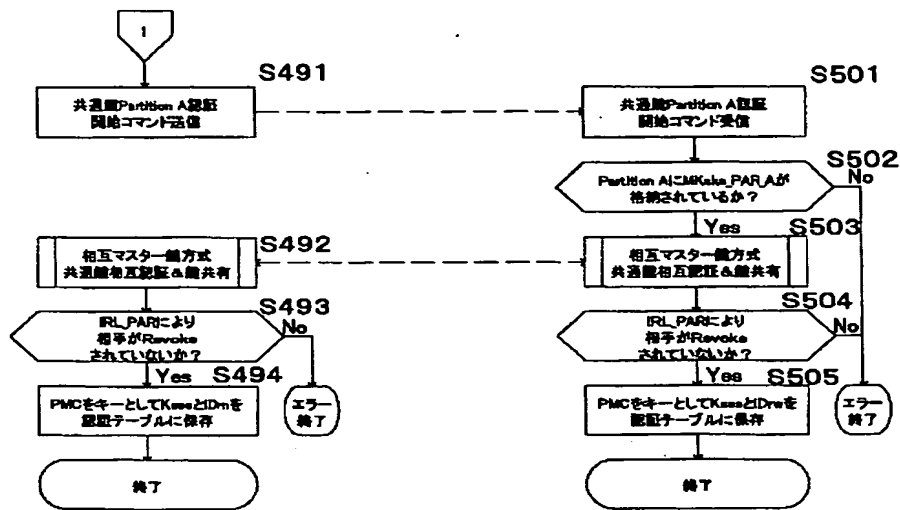


【図55】



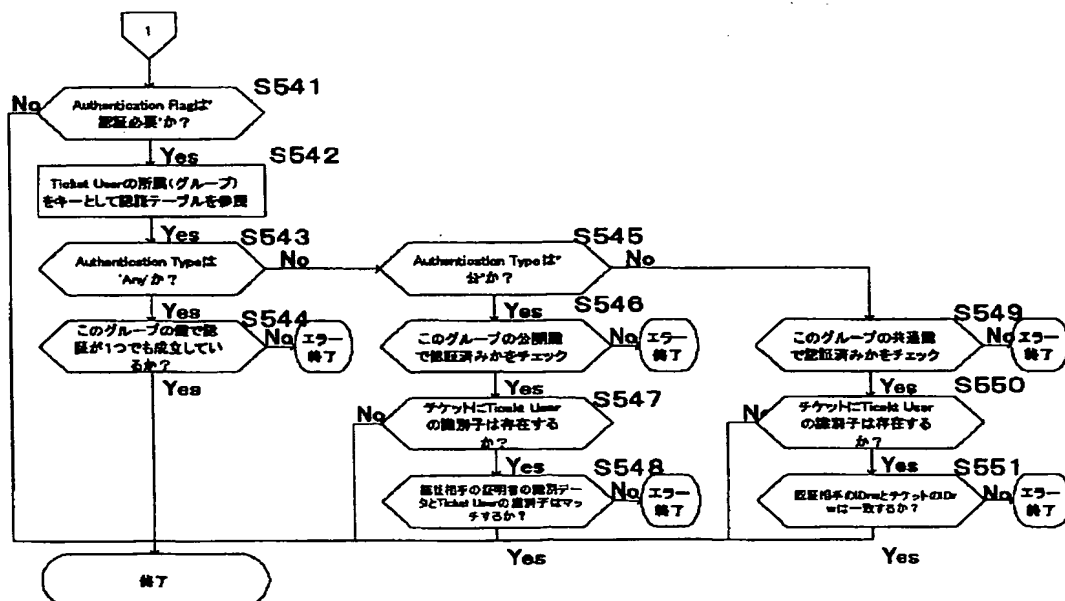
Partition認証

【図56】



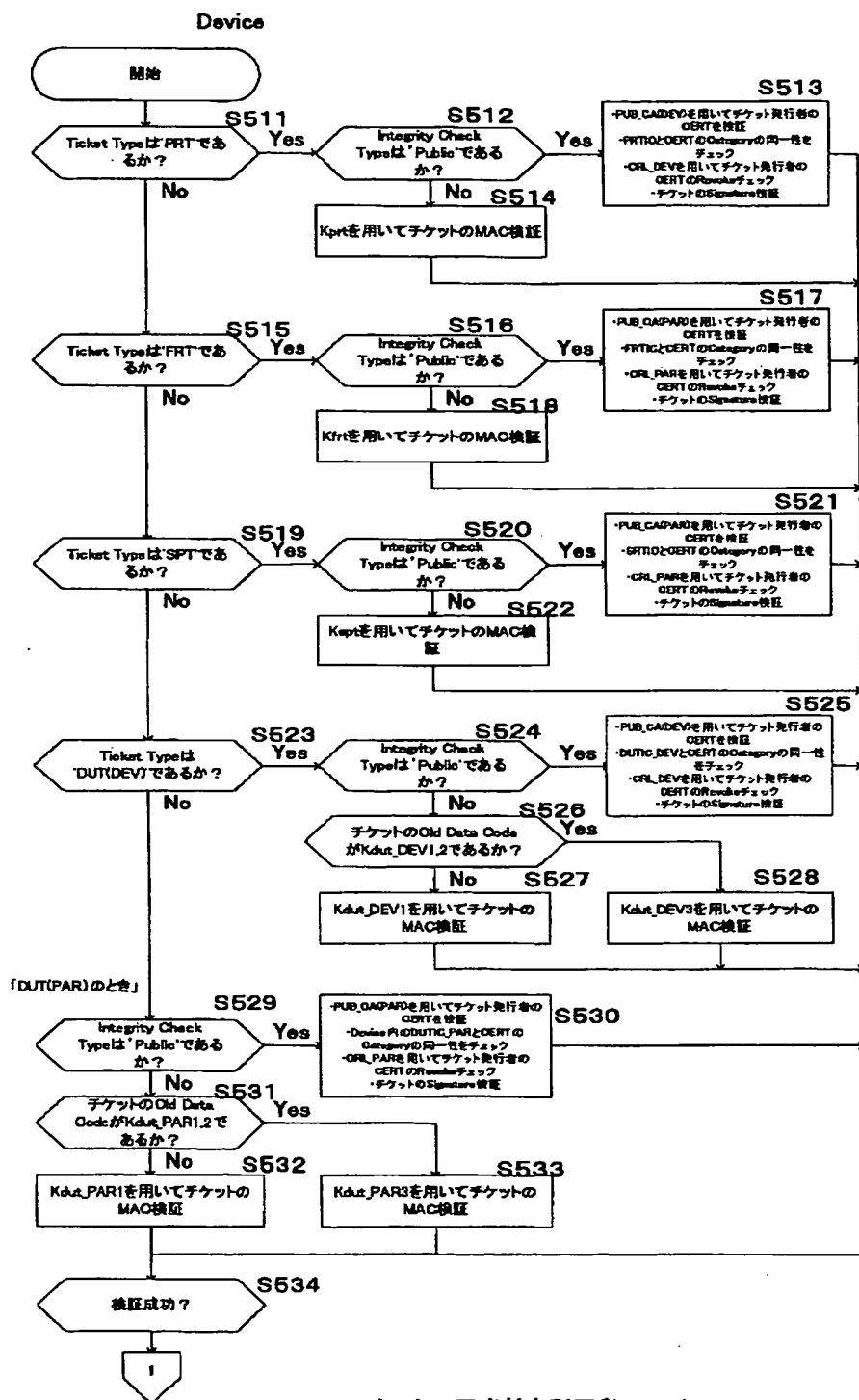
Partition認証(続き)

【図58】



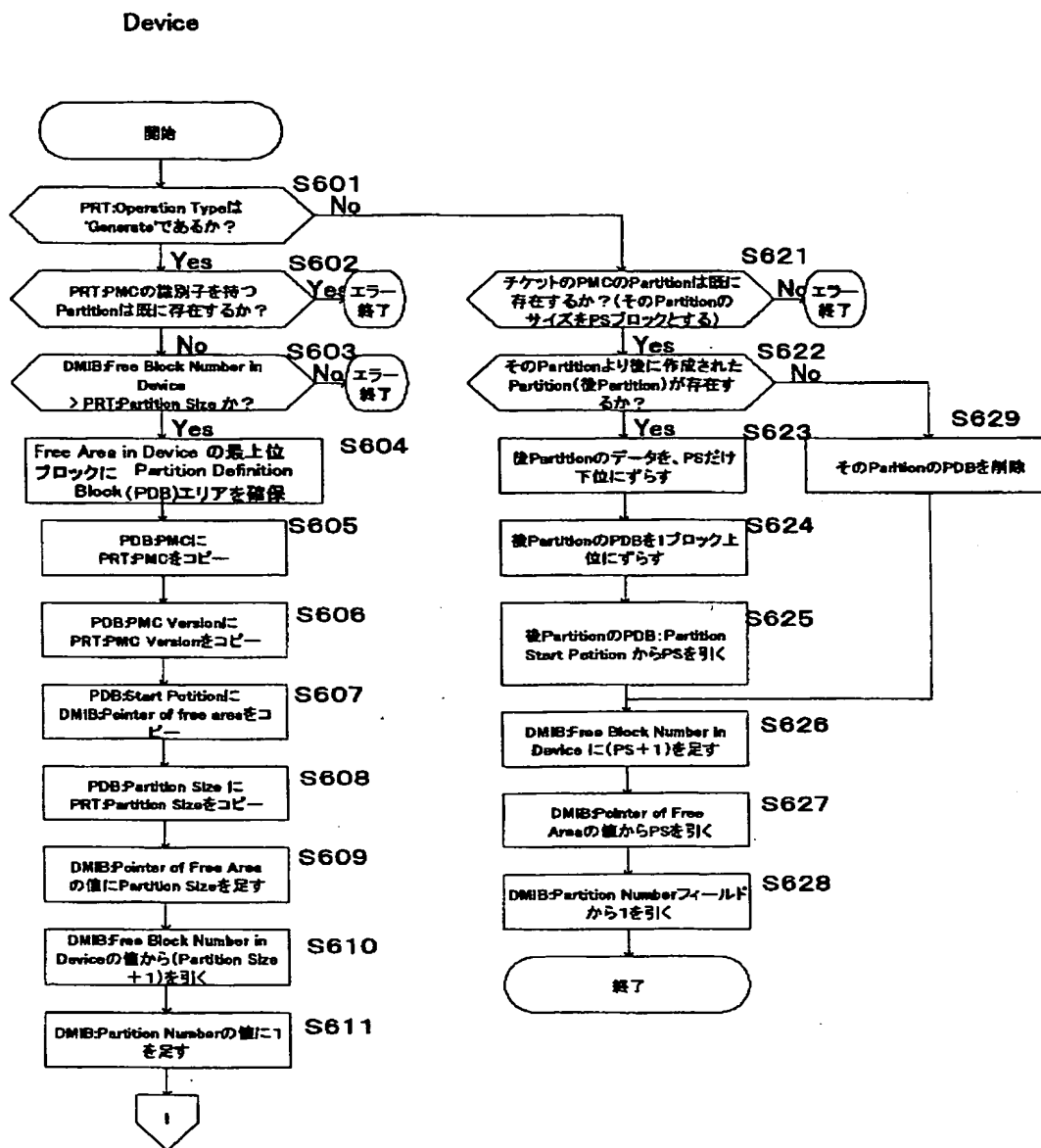
チケットの正当性と利用者チェック(続き)

【図57】



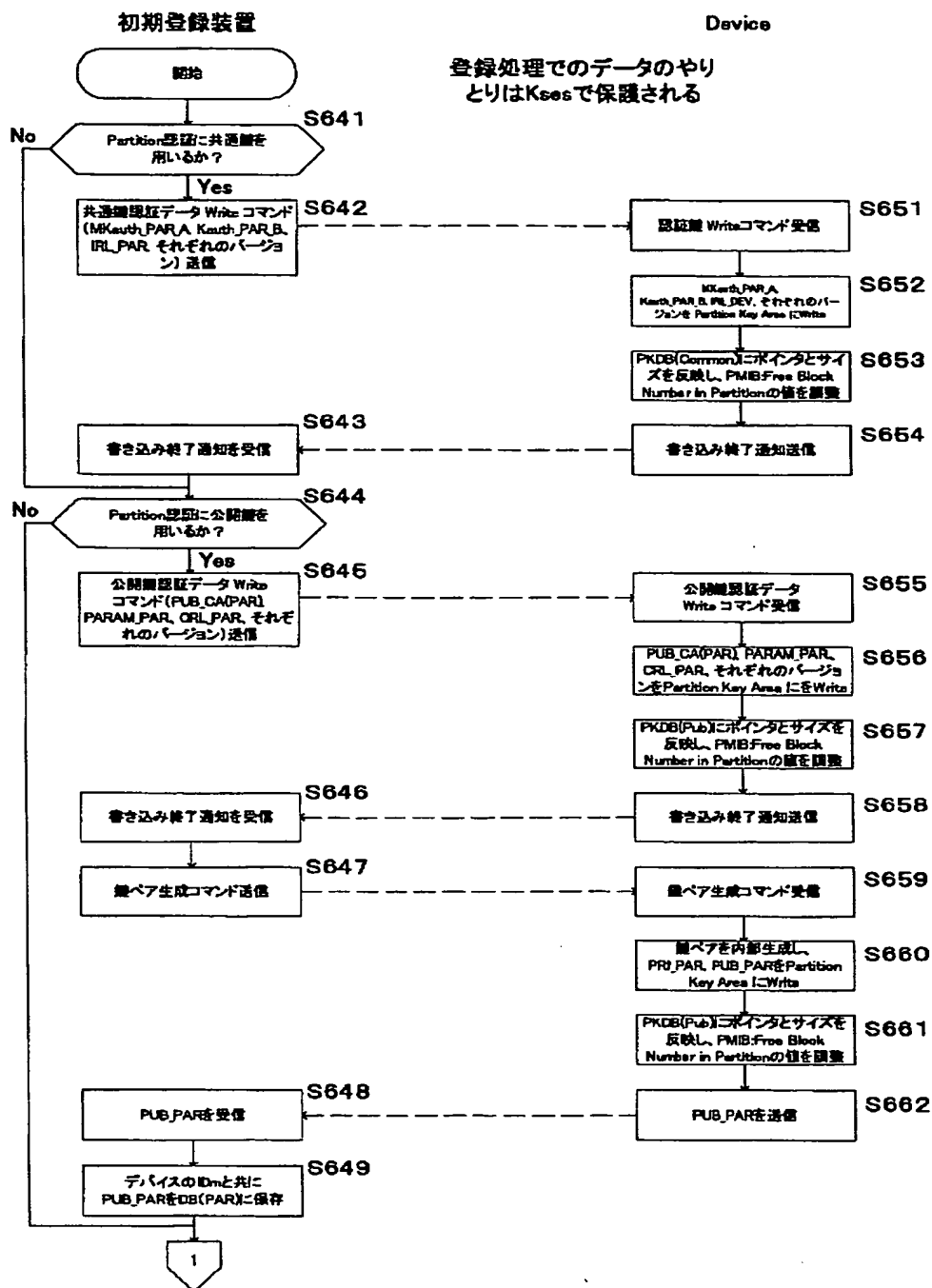


【図60】



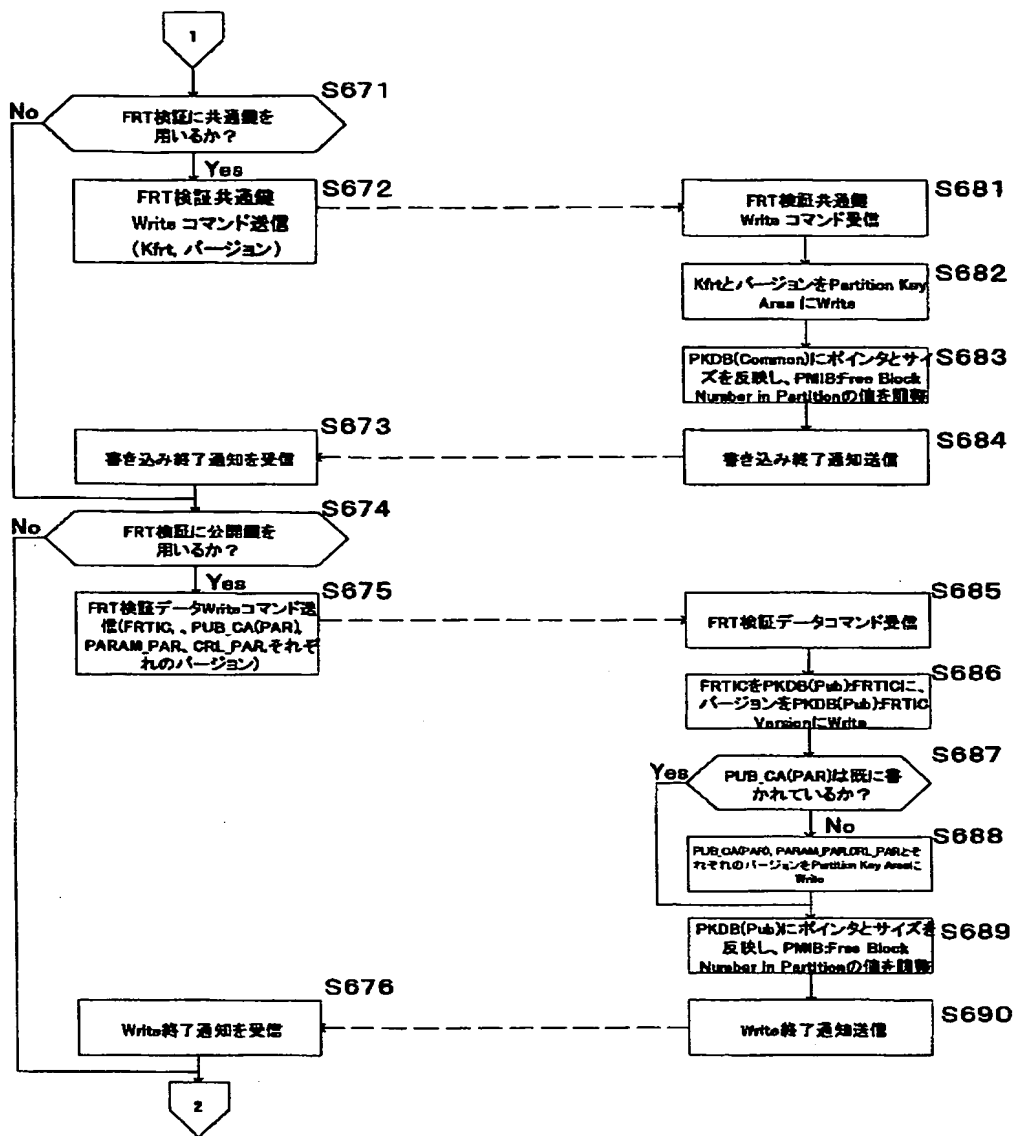
Partition作成・削除操作

登録処理でのデータのやりとりはKsesで保護される



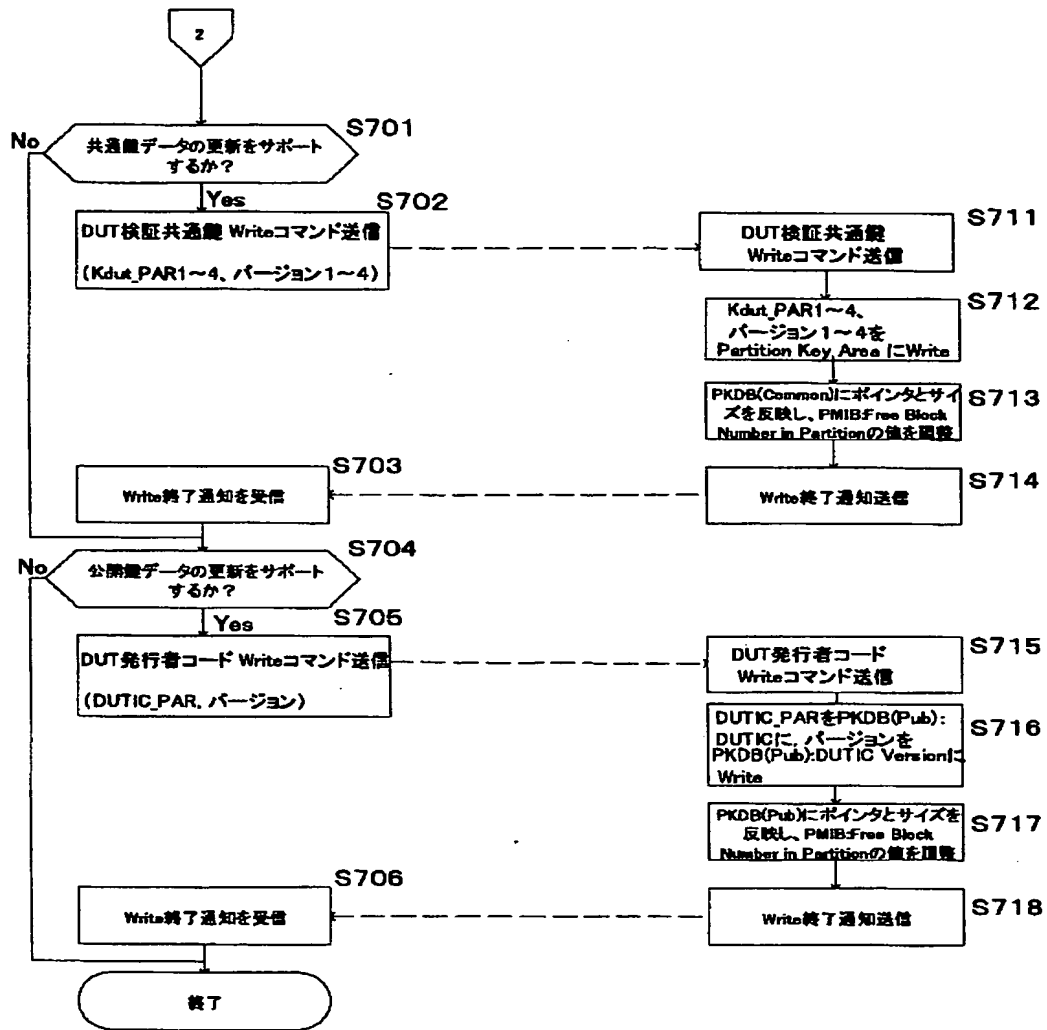
## Partition初期登録

【図63】



Partition初期登録(続き)

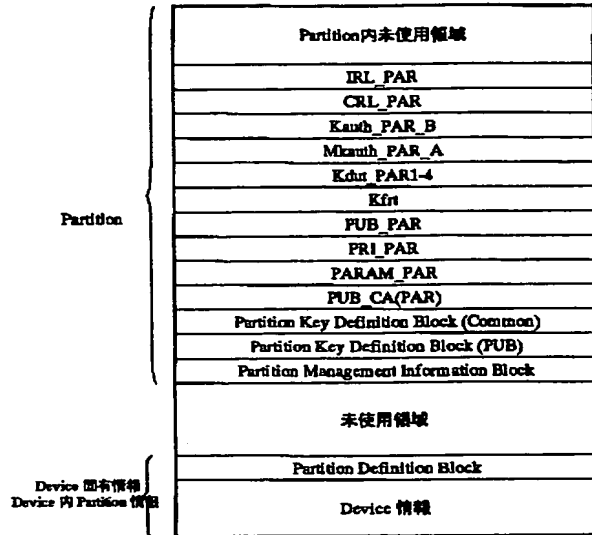
【図64】



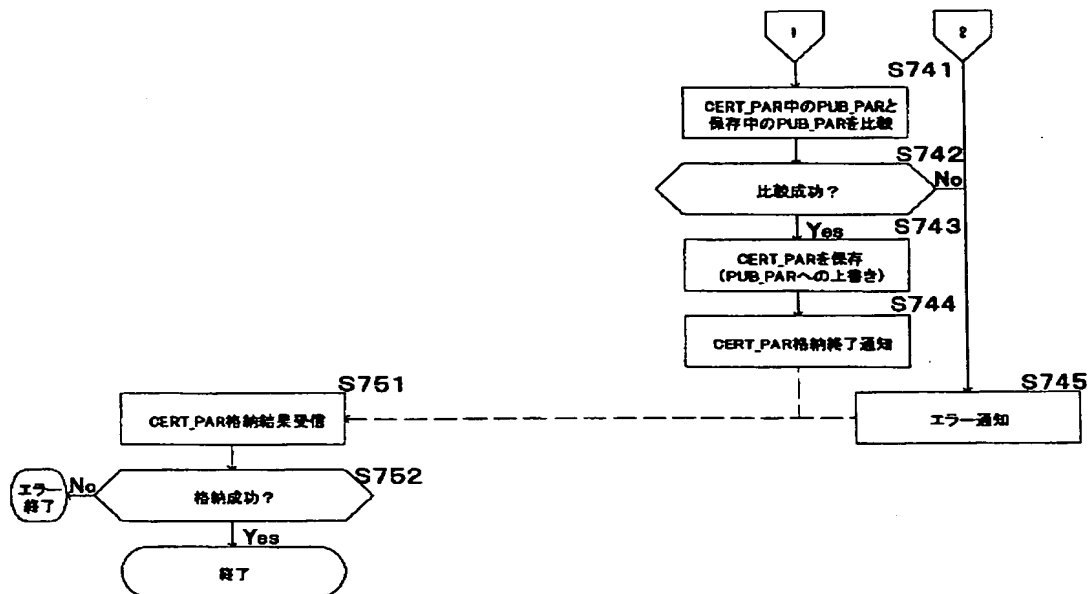
Partition初期登録(続き)

【図65】

## Partition生成後の状態

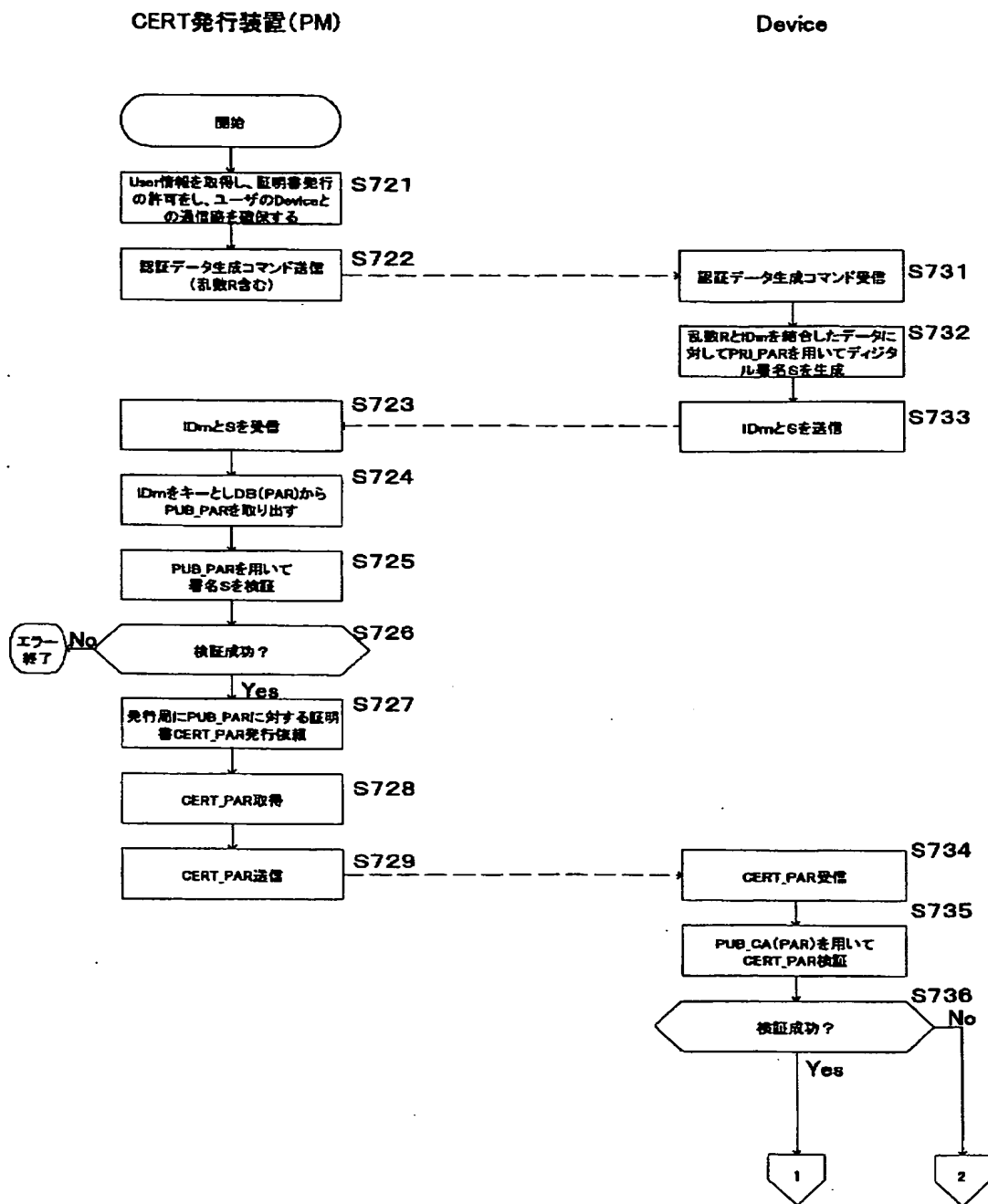


【図67】



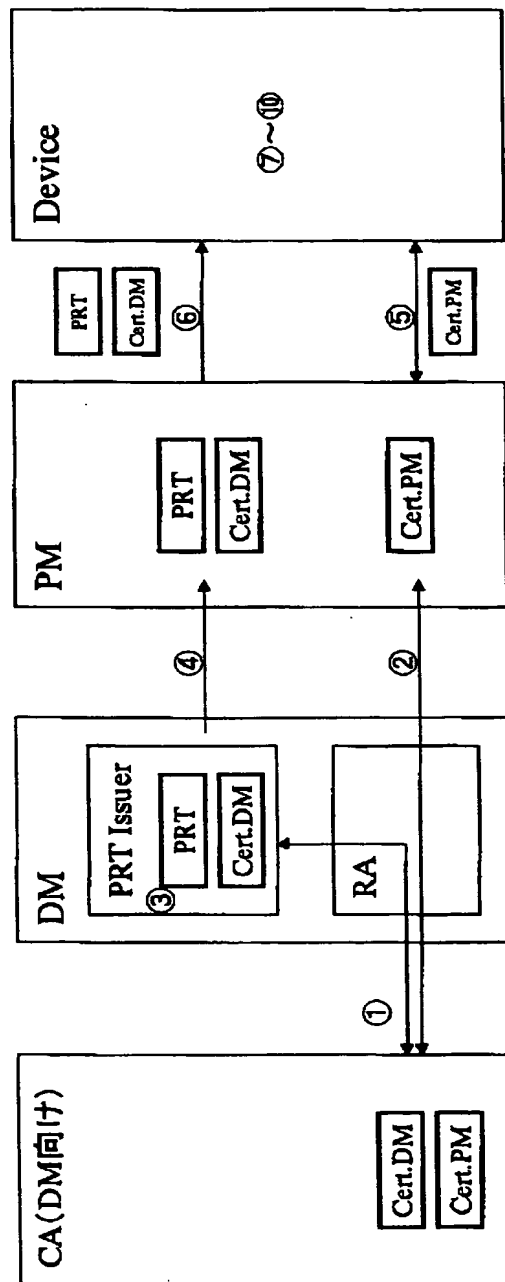
CERT発行(PM)(続き)

【図 6 6】



CERT発行(PM)

Partition生成手順 (Authentication Type: 公開鍵, PRT:公開鍵)



【図 6 8】

- ①DM(Device Manager)用の公開鍵証明書発行
- ②PM(Partition Manager)用の公開鍵証明書の発行
- ③PRT(Partition Registration Ticket)の生成
- ④PRT及びDMのCertificateの供給
  - PRTには、検証値(公開鍵)が付いている
- ⑤PMとDeviceの間の相互認証(公開鍵)
- ⑥PRT及びDMのCertificateの送信
  - PRTの検証
  - PRT生成者の検証、PRT使用者の検証
- ⑦Partitionの生成
- ⑧鍵データ書き込み
- ⑨Public Keyの読み出し(作成したPartitionでは公開鍵認証を用いる場合)
- ⑩Certificateの発行(作成したPartitionでは公開鍵認証を用いる場合)

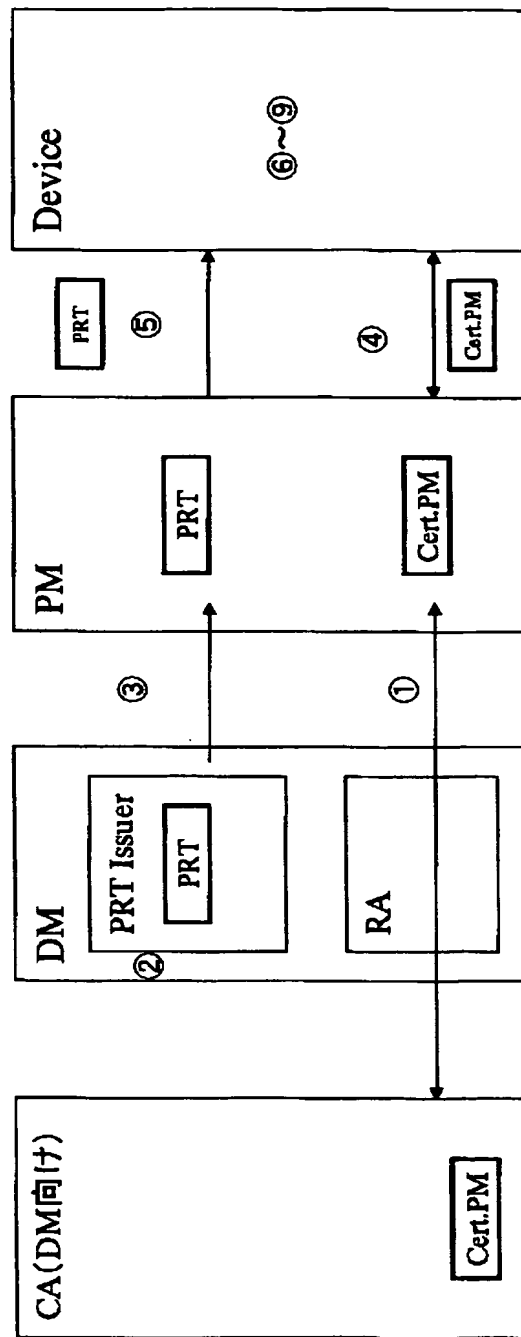
PRT例

PRT Version Number
PRT Size
Authentication Type=公
Category:PRT User
Partition Manager Code
Partition Size
Signature

Certificate例

Certificate Version
SN
DN:Device Manager
...
DM Public Key
Group:DMC
Category:PRT Issuer
Signature

Partition 生成手順 (Authentication Type: 公開鍵, PRT: 共通鍵)



【図 6 9】

①PM(Partition Manager)用の公開鍵証明書の発行

②PRT(Partition Registration Ticket)の生成

③PRTの供給

→PRTには、検証値(共通鍵)が付いている

④PMとDeviceの間の相互認証(公開鍵)

⑤PRTの送信

→PRTの検証

→PRT生成者の検証、PRT使用者の検証

⑥Partitionの生成

⑦鍵データ書き込み

⑧Public Keyの読み出し

(作成したPartitionでは公開鍵認証を用いる場合)

⑨Certificateの発行(作成したPartitionでは公開鍵認証を用いる場合)

PRT例

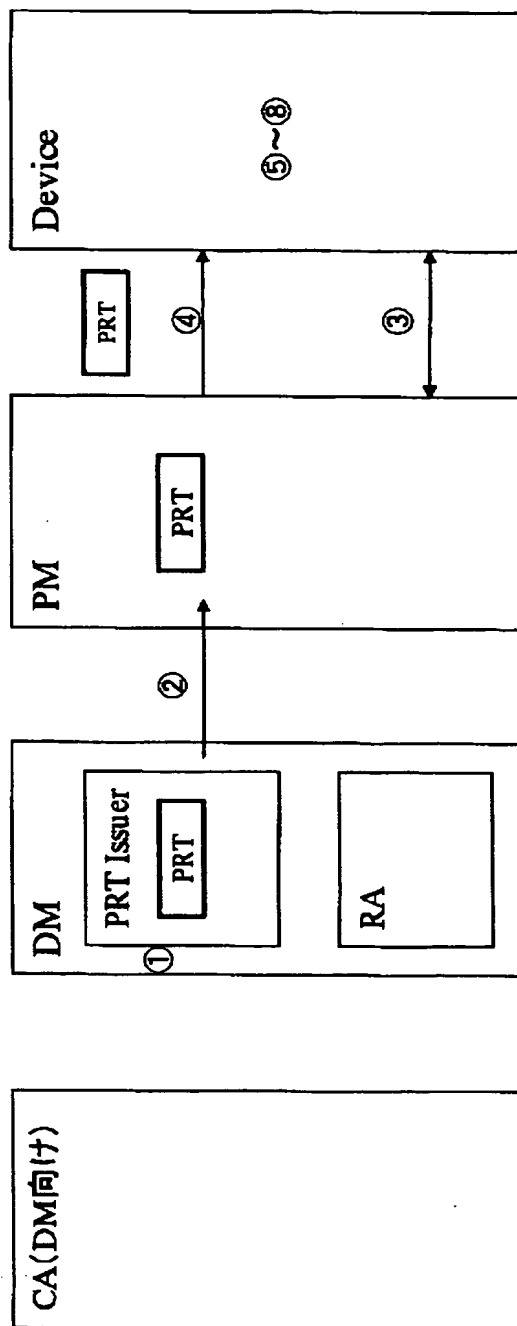
PRT Version Number
PRT Size
Authentication Type=公
Category:PRT User
Partition Manager Code
Partition Size
MAC

Certificate例

Certificate Version
SN
DN:Device Manager
...
DM Public Key
Group:DMC
Category:PRT Issuer
Signature



Partition生成手順 (Authentication Type: 共通鍵, PRT:共通鍵)



【図70】

- ①PRT (Partition Registration Ticket) の生成
- ②PRT の供給  
→PRTには、検証値(共通鍵)が付いている
- ③PMとDeviceの間の相互認証(共通鍵)
- ④PRTの送信  
→PRTの検証  
→PRT生成者の検証、PRT使用者の検証
- ⑤Partitionの生成
- ⑥鍵データ書き込み
- ⑦Public Keyの読み出し  
(作成したPartitionでは公開鍵認証を用いる場合)
- ⑧Certificateの発行(作成したPartitionでは公開鍵認証を用いる場合)

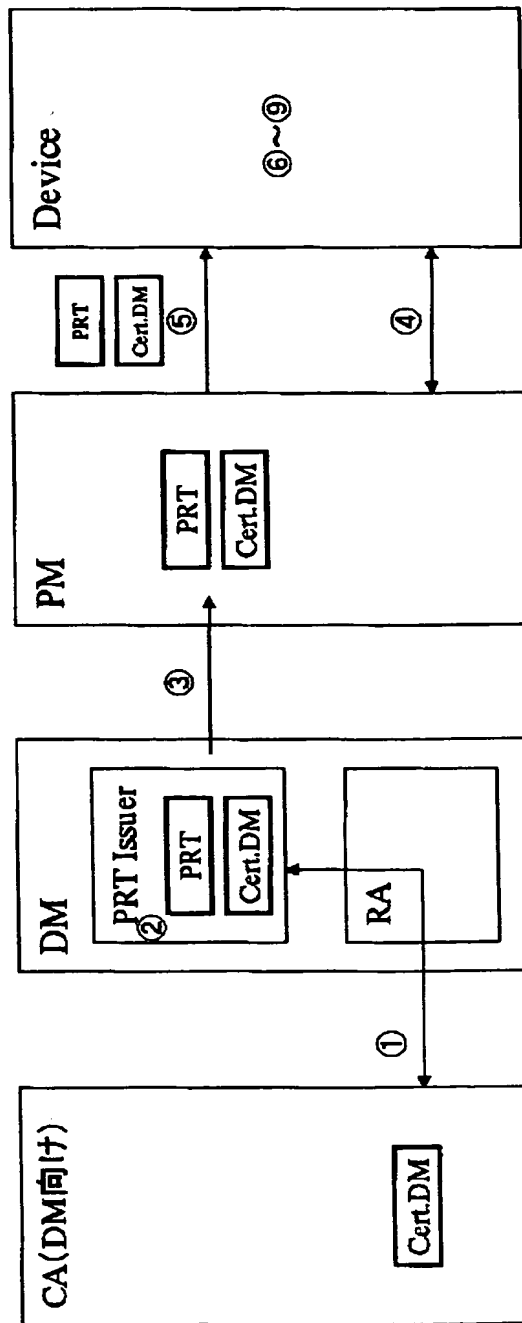
PRT例

PRT Version Number
PRT Size
Authentication Type=共
Category:PRT User
Partition Manager Code
Partition Size
MAC

Certificate例

Certificate Version
SN
DN:Device Manager
...
DM Public Key
Group:DMC
Category:PRT Issuer
Signature

## Partition 生成手順 (Authentication Type: 共通鍵, PRT: 公開鍵)



【図 7 1】

①DM(Device Manager)用の公開鍵証明書の発行

②PRT(Partition Registration Ticket)の生成

③PRT及びDMのCertificateの供給

→PRTには、検証値(公開鍵)が付いている

④PMとDeviceの間の相互認証(共通鍵)

⑤PRT及びDMのCertificateの送信

→PRTの検証

→PRT生成者の検証、PRT使用者の検証

⑥Partitionの生成

⑦鍵データ書き込み

⑧Public Keyの読み出し

(作成したPartitionでは公開鍵認証を用いる場合)

⑨Certificateの発行(作成したPartitionでは公開鍵認証を用いる場合)

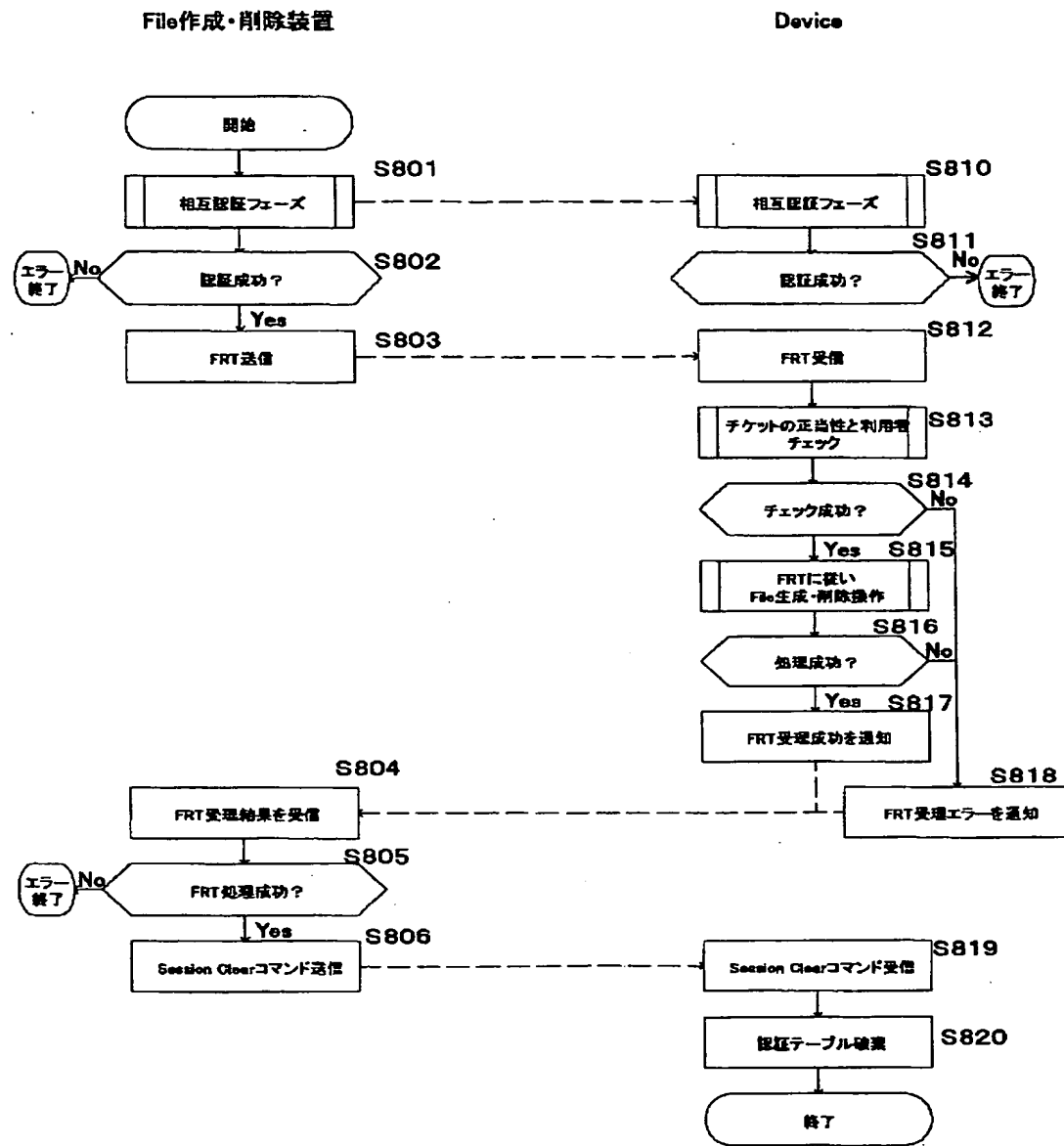
## PRT例

PRT Version Number
PRT Size
Authentication Type-共
Category:PRT User
Partition Manager Code
Partition Size
Signature

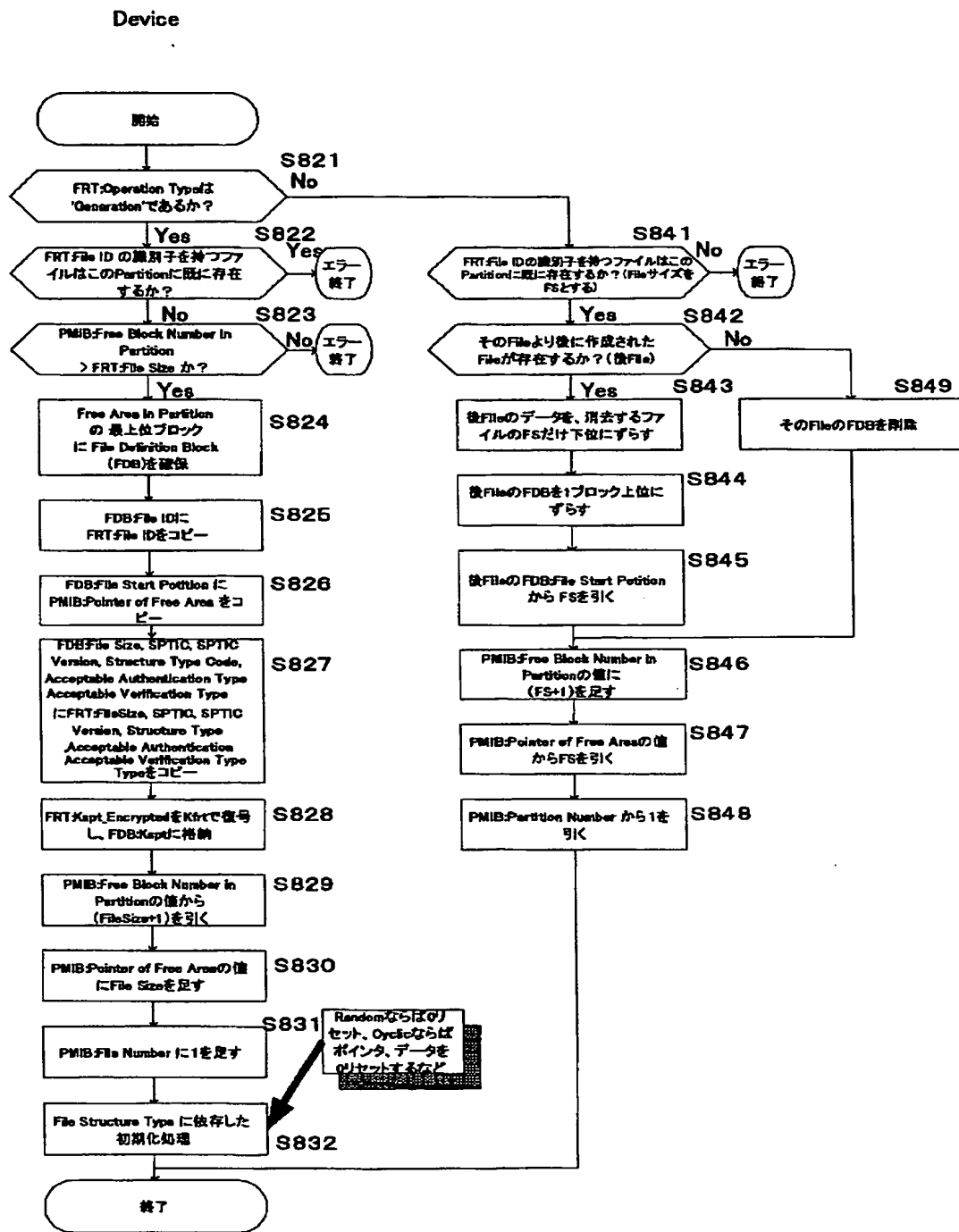
## Certificate例

Certificate Version
SN
DN:Device Manager
...
DM Public Key
Group:DMC
Category:PRT Issuer
Signature

【図72】



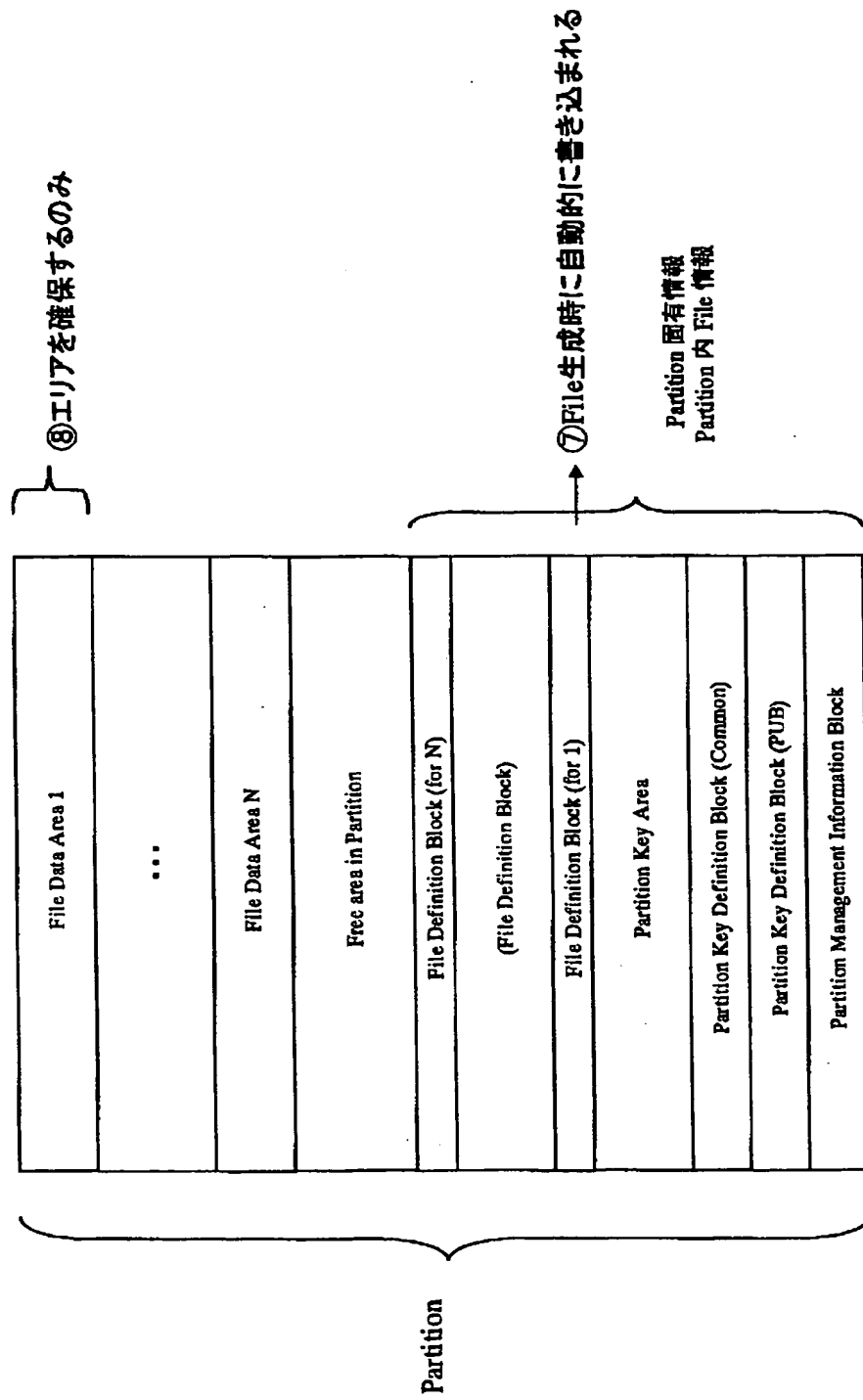
【図73】



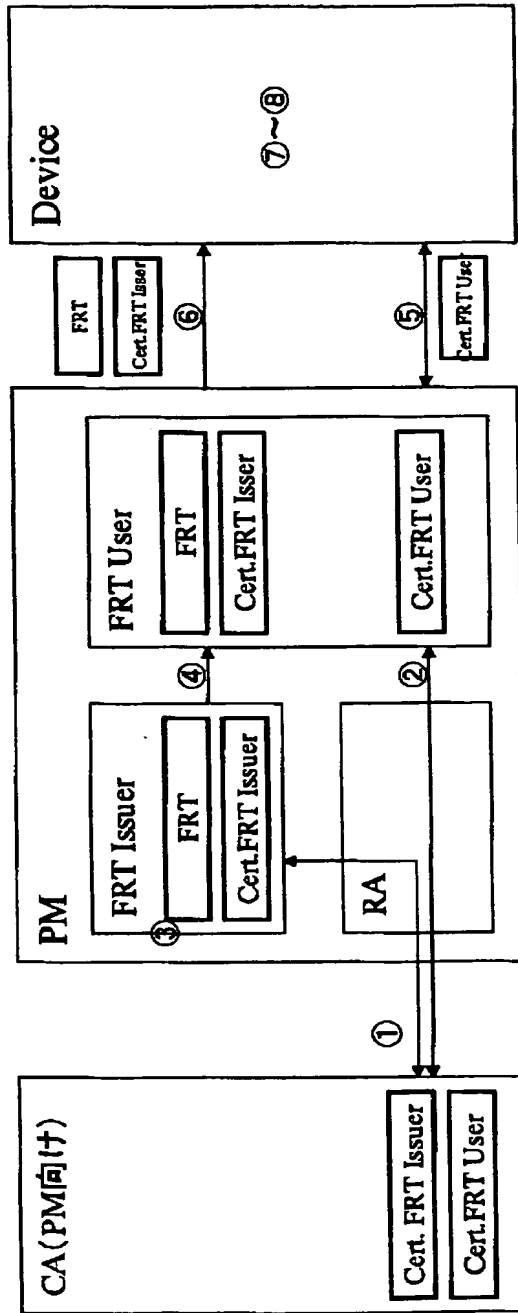
File作成・削除操作

【図 7 4】

File 生成後の状態



File生成手順 (Authentication Type: 公開鍵, FRT:公開鍵)



【図 7 5】

- ① FRT Issuer用の公開鍵証明書の発行
- ② FRT User用の公開鍵証明書の発行
- ③ FRT (File Registration Ticket) の生成
- ④ FRT及びFRT Issuer用のCertificateの供給  
→ FRTには、検証値(公開鍵)が付いている
- ⑤ FRT IssuerとDeviceの間の相互認証(公開鍵)
- ⑥ FRT及びFRT Issuer用のCertificateの送付  
→ FRTの検証  
→ FRT生成者の検証、FRT使用者の検証
- ⑦ File Definition Block (SPT Issuer CategoryとKspiを登録
- ⑧ File Data Areaを確保

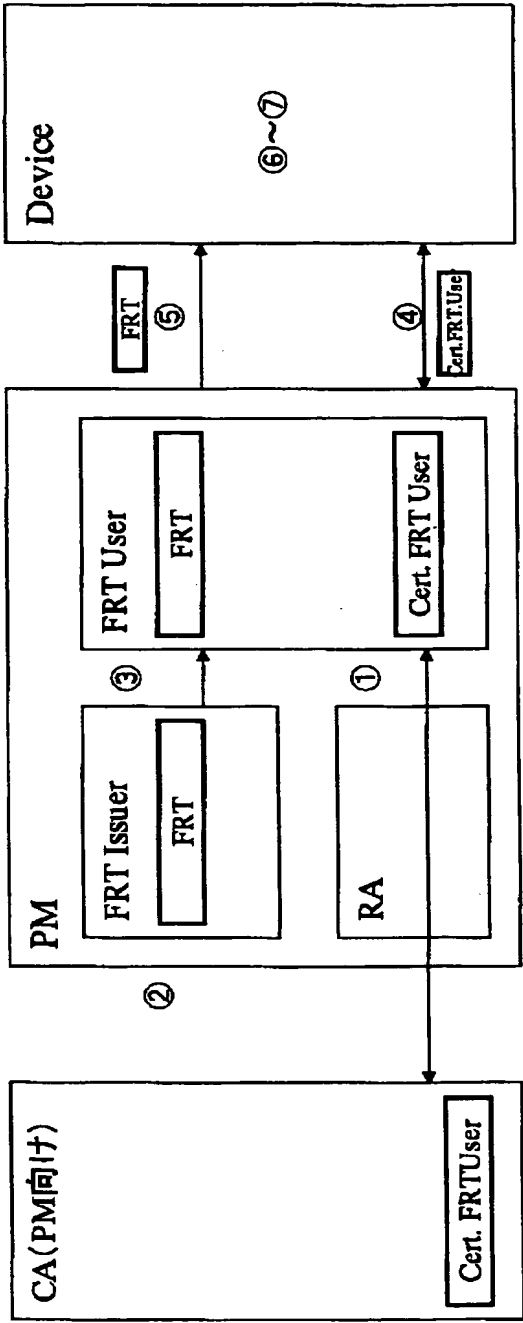
FRT例

...
Authentication Type=公
Category:FRT User
Partition Manager Code
File Information
Category:SPT Issuer
Kfnt(Kspi)
Signature

Certificate例

Certificate Version
SN
DN: Partition Manager
...
PM Public Key
Group: PMC
Category: FRT Issuer
Signature

File生成手順 (Authentication Type: 公開鍵, FRT: 共通鍵)



【図 7 6】

- ① FRT User用の公開鍵証明書の発行
- ② FRT (File Registration Ticket) の生成
- ③ FRTの供給
  - FRTには、検証値(共通鍵)が付いている
- ④ FRT IssuerとDeviceの間の相互認証(公開鍵)
- ⑤ FRTの送信
  - FRTの検証
  - FRT生成者の検証、FRT使用者の検証
- ⑥ File Definition Block(SPT Issuer CategoryとKspiを登録
- ⑦ File Data Areaを確保

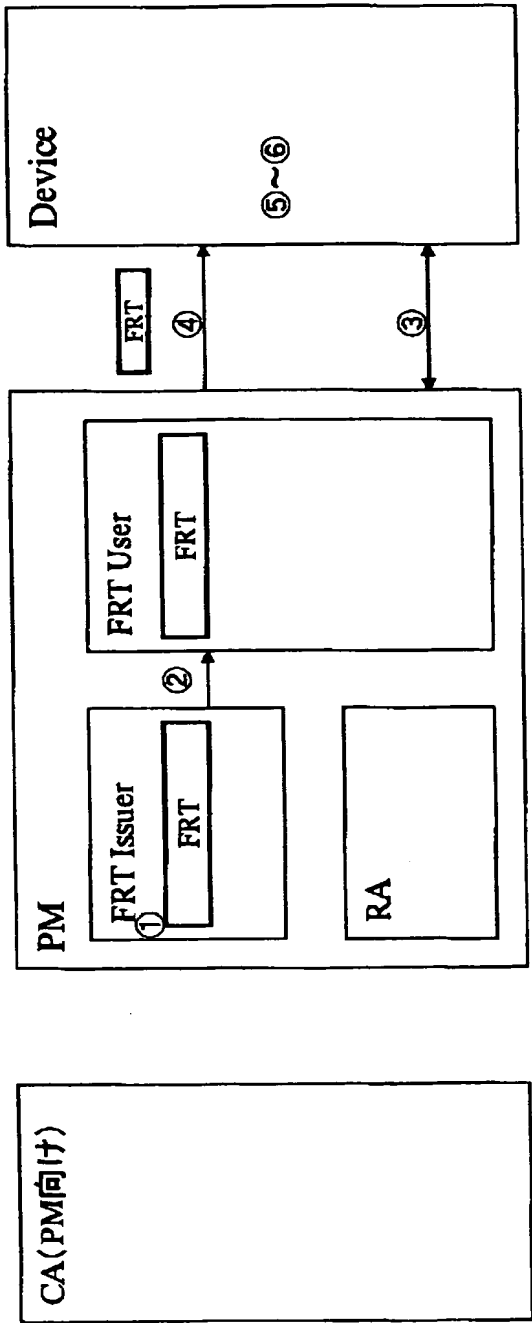
FRT例

...
Authentication Type=公
Category:FRT User
Partition Manager Code
File Information
Category:SPT Issuer
Kspi(Kspi)
MAC

Certificate例

Certificate Version
SN
DN:Partition Manager
...
PM Public Key
Group:PMC
Category:FRT Issuer
Signature

File生成手順 (Authentication Type: 共通鍵, FRT: 共通鍵)



【図 77】

① FRT (File Registration Ticket) の生成

② FRT の供給

→ FRT には、検証値 (共通鍵) が付いている

③ FRT Issuer と Device の間の相互認証 (共通鍵)

④ FRT の送信

→ FRT の検証

→ FRT 生成者の検証、FRT 使用者の検証

⑤ File Definition Block (SPT Issuer Category と Kspi を登録)

⑥ File Data Area を確保

FRT 例

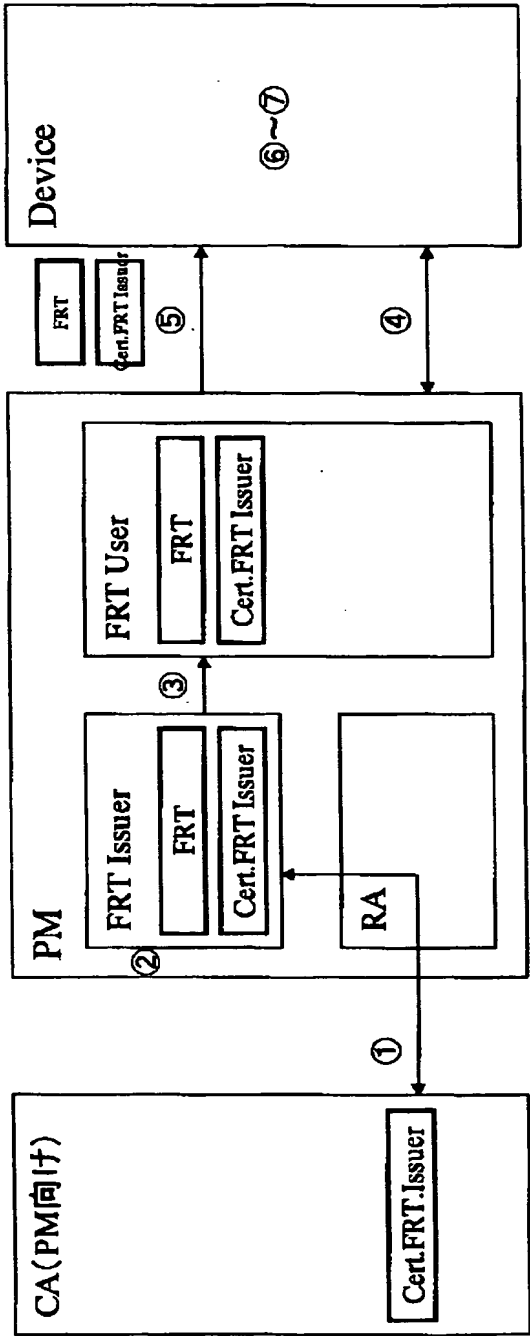
...
Authentication Type=共
Category:FRT User
Partition Manager Code
File Information
Category:SPT Issuer
Kspi(Kspi)
MAC

Certificate例

Certificate Version
SN
DN:Partition Manager
...
PM Public Key
Group:PMC Category:FRT Issuer
Signature



File生成手順 (Authentication Type: 共通鍵, FRT:公開鍵)



【図78】

- ①FRT Issuer用の公開鍵証明書の発行  
②FRT(File Registration Ticket)の生成  
③FRT及びFRT Issuer用のCertificateの供給  
→FRTIには、検証値(公開鍵)が付いている  
④FRT IssuerとDeviceの間の相互認証(共通鍵)  
⑤FRT及びFRT Issuer用のCertificateの送信  
→FRTの検証  
→FRT生成者の検証、FRT使用者の検証  
⑥File Definition Block( SPT Issuer CategoryとKspIを登録  
⑦File Data Areaを確保

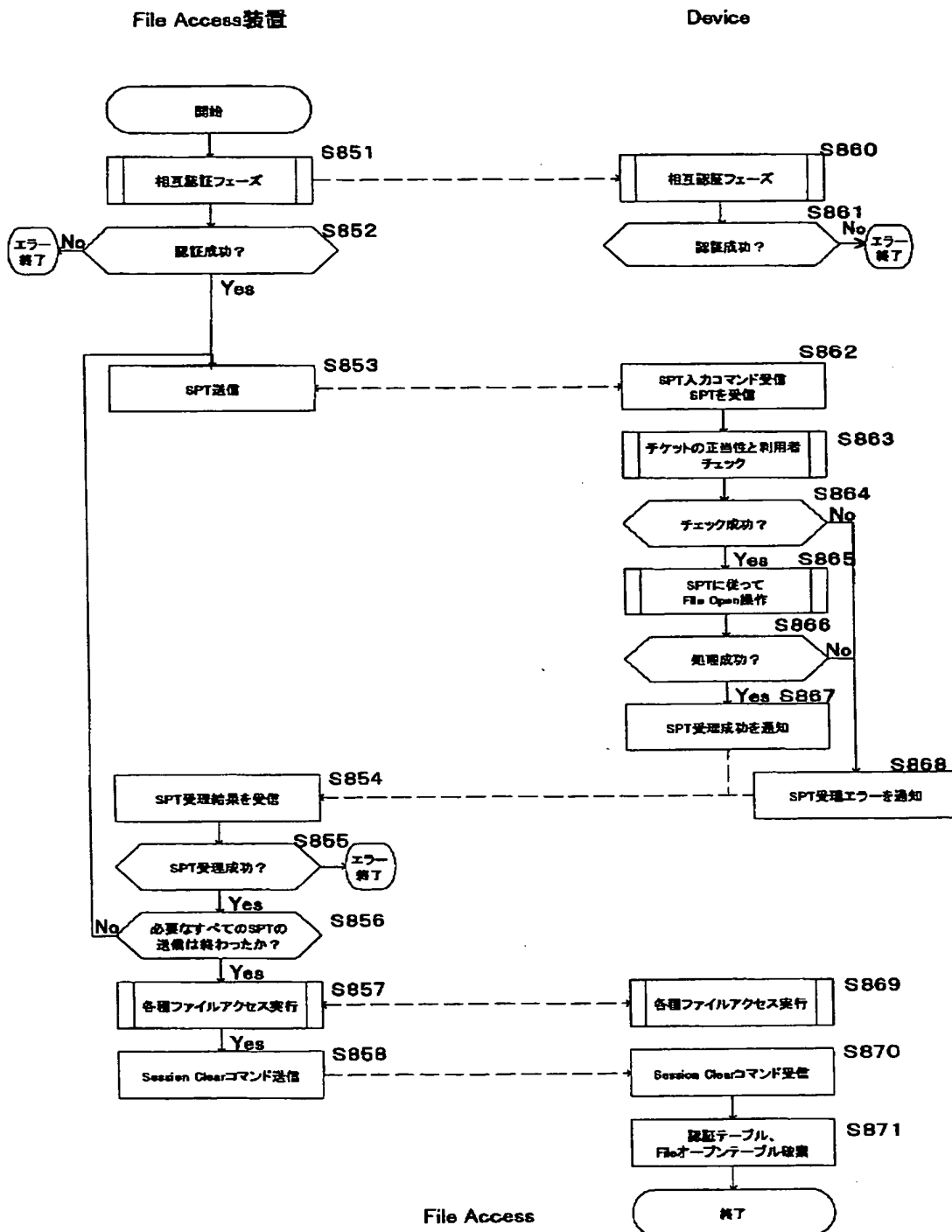
FRT例

...
Authentication Type=共
Category:FRT User
Partition Manager Code
File Information
Category:SPT Issuer
Krr(KspI)
Signature

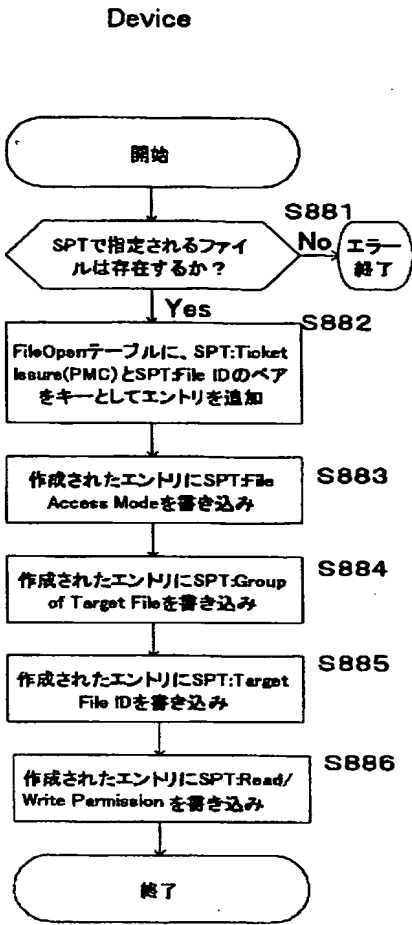
Certificate例

Certificate Version
SN
DN:Partition Manager
...
PM Public Key
Group:PMC
Category:FRT Issuer
Signature

【図 7 9】



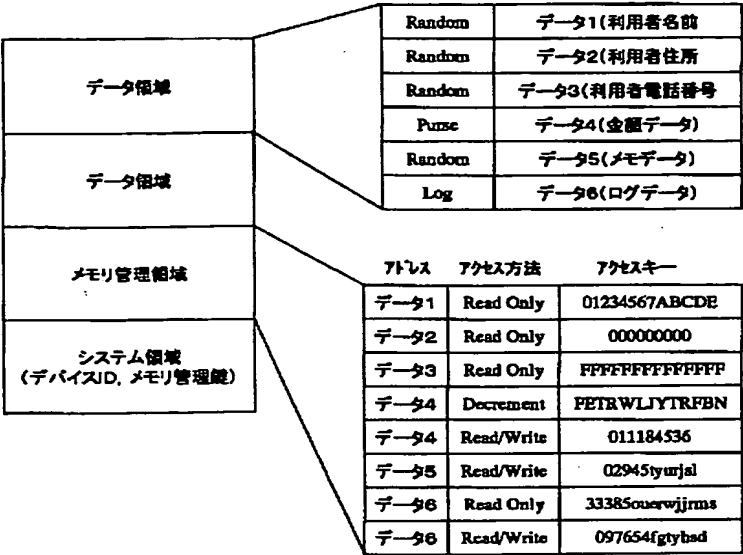
【図80】



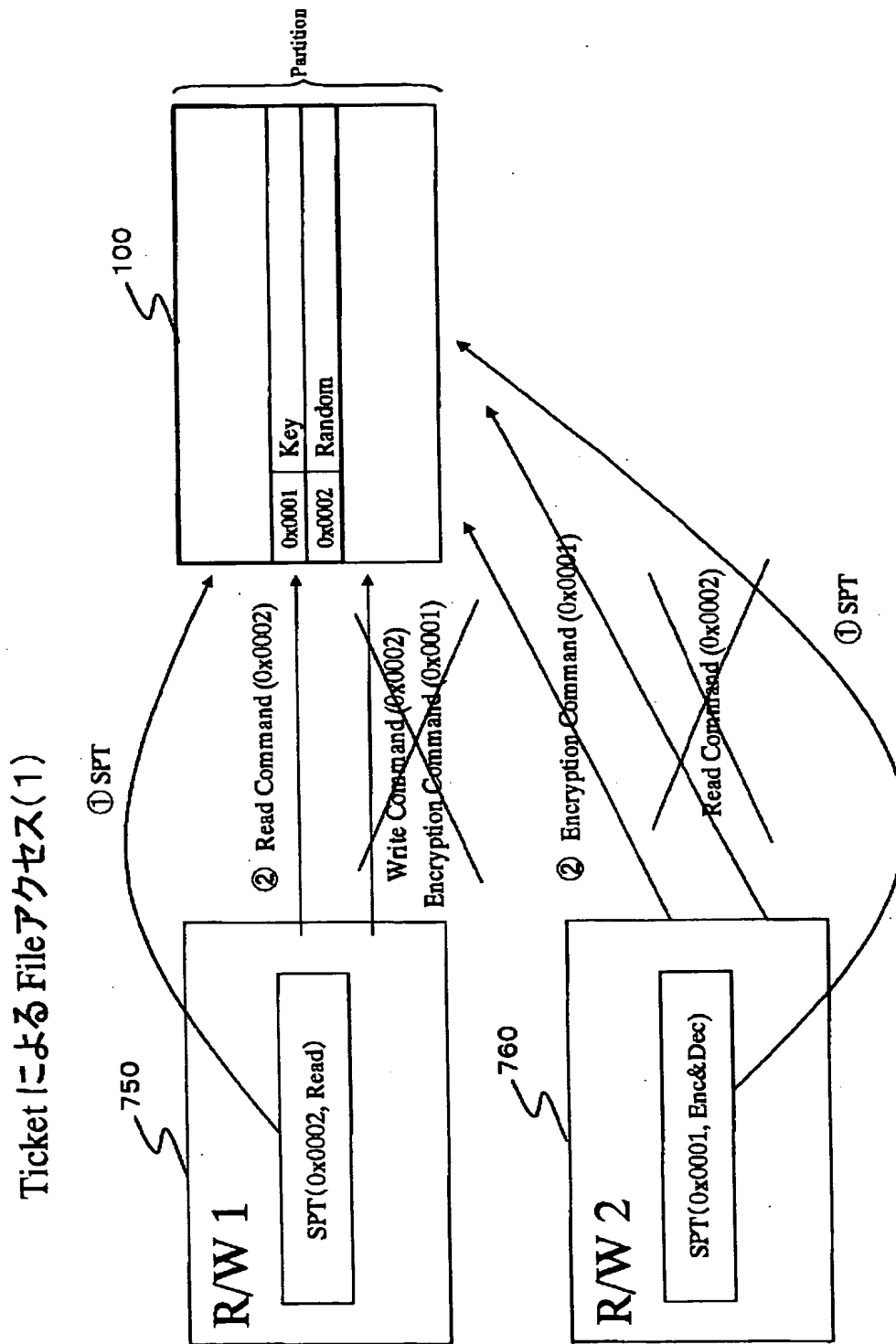
SPTに従ってFile Open操作

【図96】

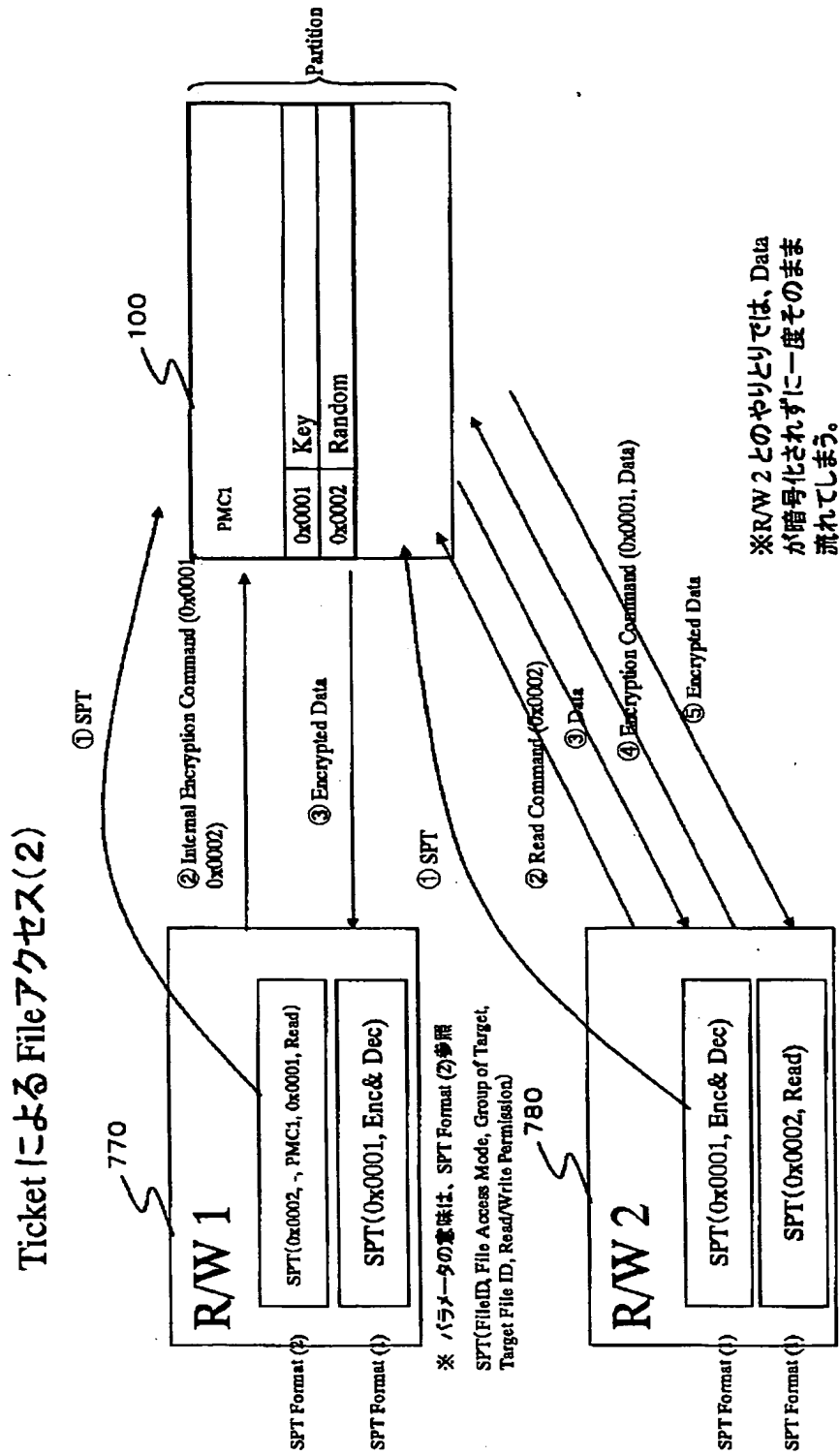
従来のメモリ構造



【図83】

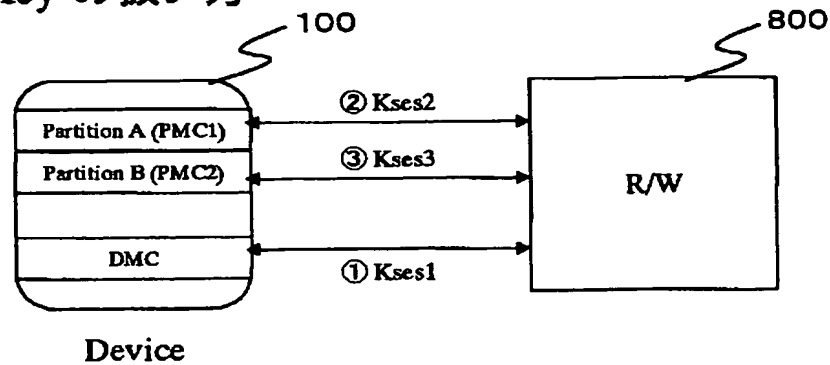


【図84】



【図85】

## 複数 Session Key の扱い方



$$1. \text{ Session Key} = \text{Kses1} \oplus \text{Kses2} \oplus \text{Kses3}$$

$$2. \text{ Session Key} = \text{Kses3} \quad \text{※一番最後のものを全体の Session Key として使う}$$

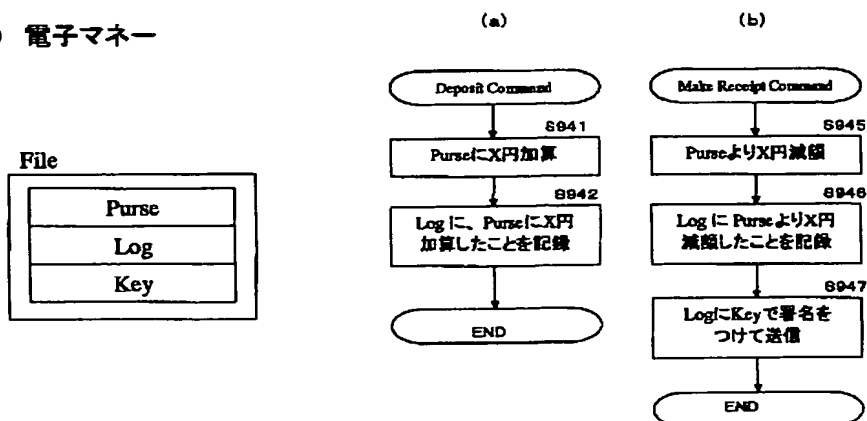


： 排他的論理和処理(8バイト単位)

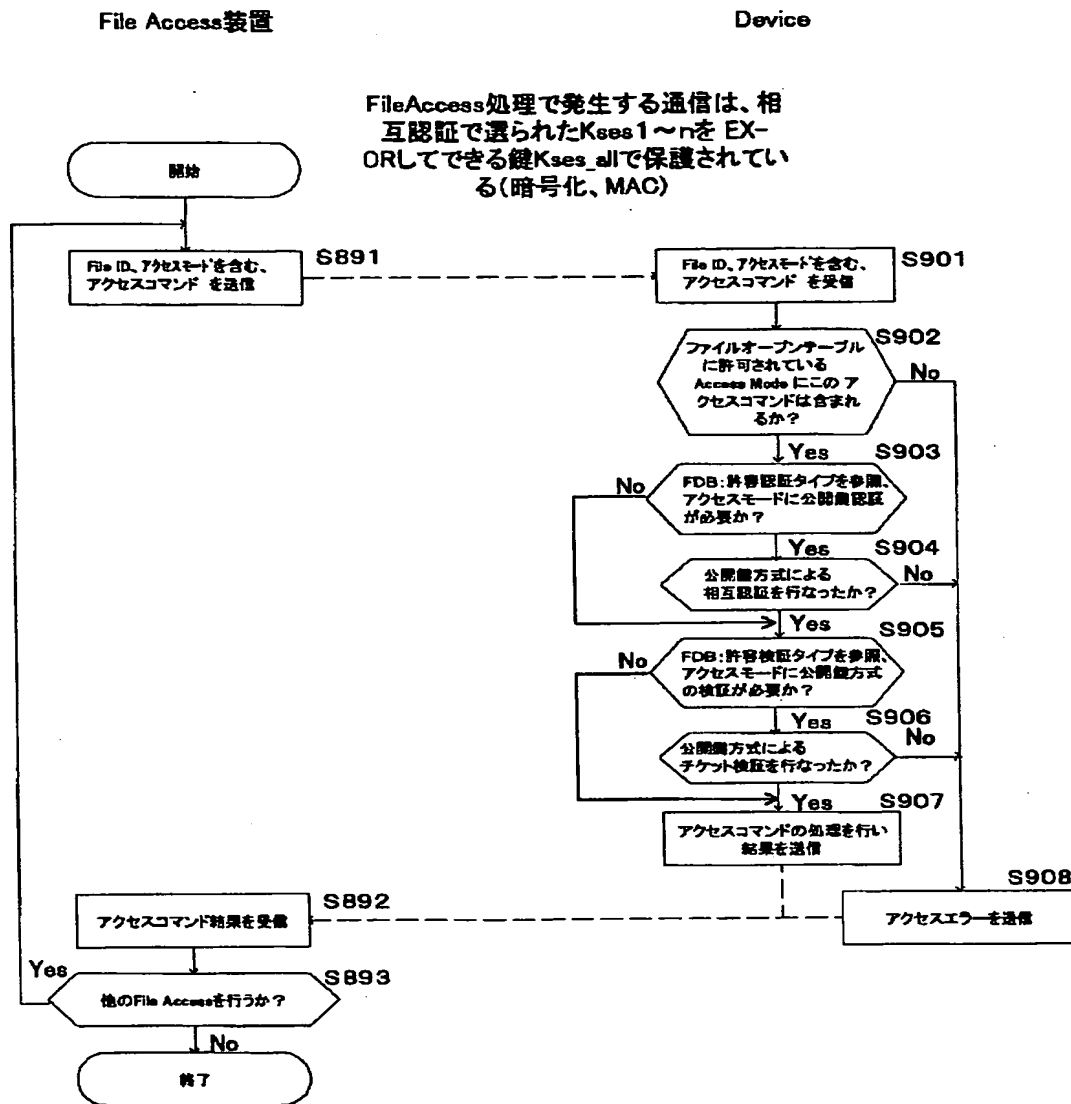
【図88】

## 複合ファイル Structure Format

例) 電子マネー

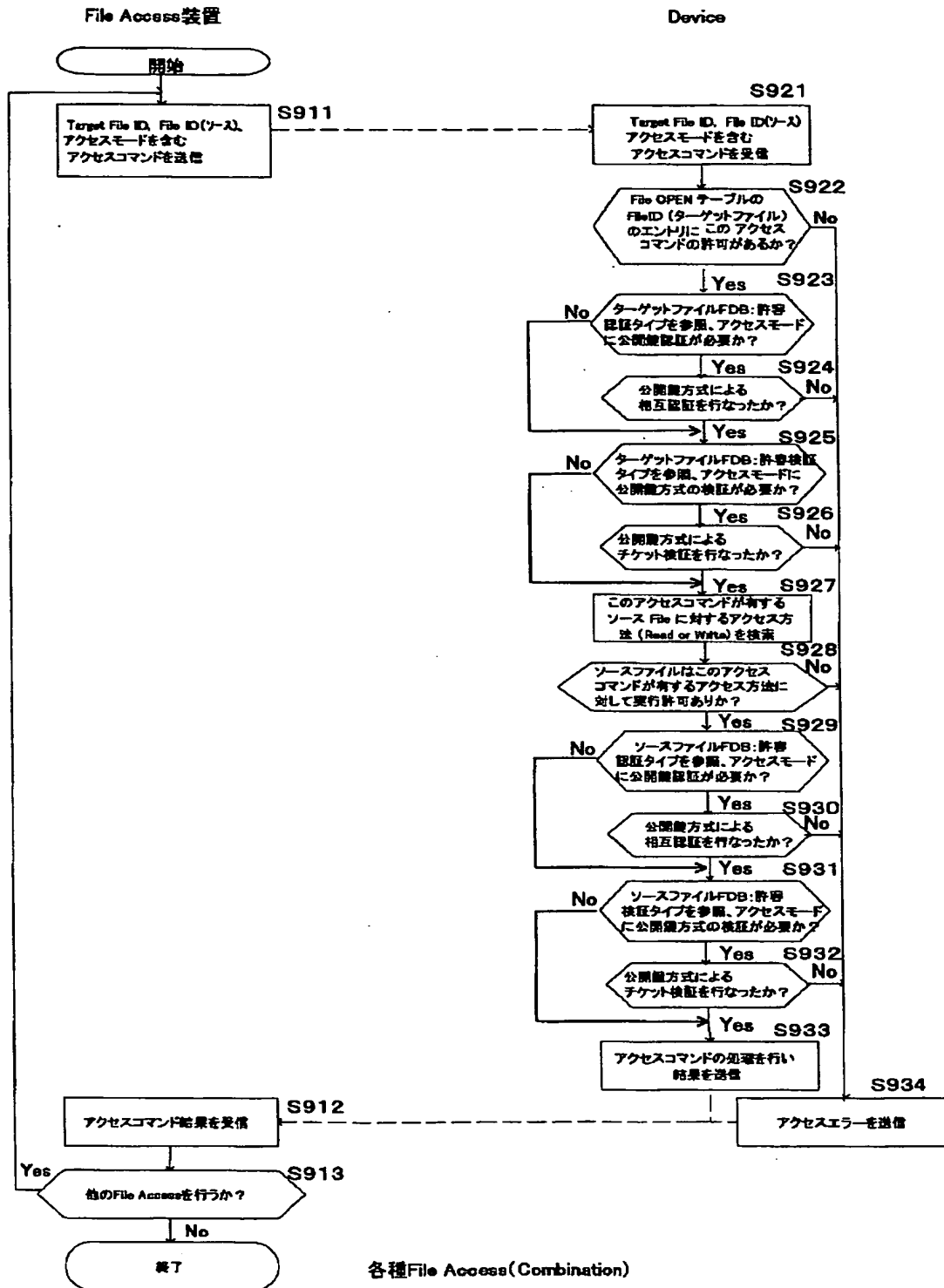


【図86】



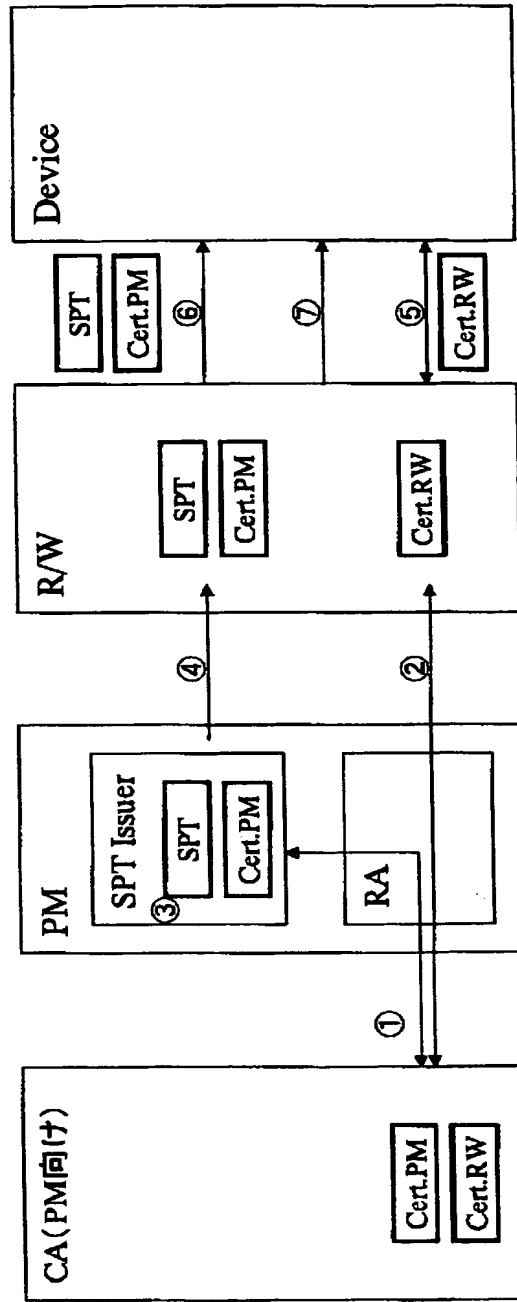
各種File Access(Normal)

【図 87】





Fileアクセス手順(Authentication Type: 公開鍵, SPT:公開鍵)



【図 8 9】

- ①PM用の公開鍵証明書の発行
- ②R/W用の公開鍵証明書の発行
- ③SPT(Service Permission Ticket)の生成
- ④SPT及びPMのCertificateの供給
  - SPTには、検証値(公開鍵)が付いている
- ⑤R/WとDeviceの間の相互認証(公開鍵)
- ⑥SPT及びPMのCertificateの送信
  - SPTの検証
  - SPT生成者の検証、SPT使用者の検証
  - SPTに示すアクセス制限が解除される
- ⑦Fileへのアクセス

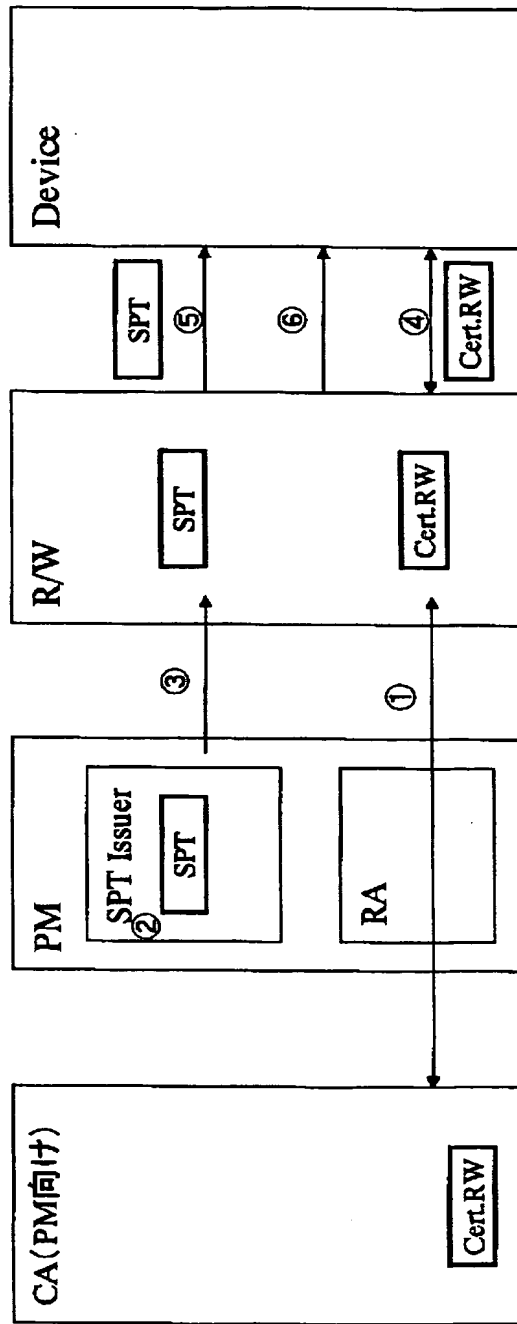
SPT例

...
Authentication Type = 公
Category:SPT User
PMCI
File ID
File Access Mode
Signature

Certificate例

Certificate Version
SN
DN:Partition Manager
...
PM1 Public Key
Group:PMC
Category:SPT Issuer
Signature

Fileアクセス手順(Authentication Type: 公開鍵, SPT: 共通鍵)



【図90】

- ①R/W用の公開鍵証明書の発行
- ②SPT(Service Permission Ticket)の生成
- ③SPTの供給
  - SPTには、検証値(共通鍵)が付いている
- ④R/WとDeviceの間の相互認証(公開鍵)
- ⑤SPTの送信
  - SPTの検証
  - SPT生成者の検証、SPT使用者の検証
  - SPTに示すアクセス制限が解除される
- ⑥Fileへのアクセス

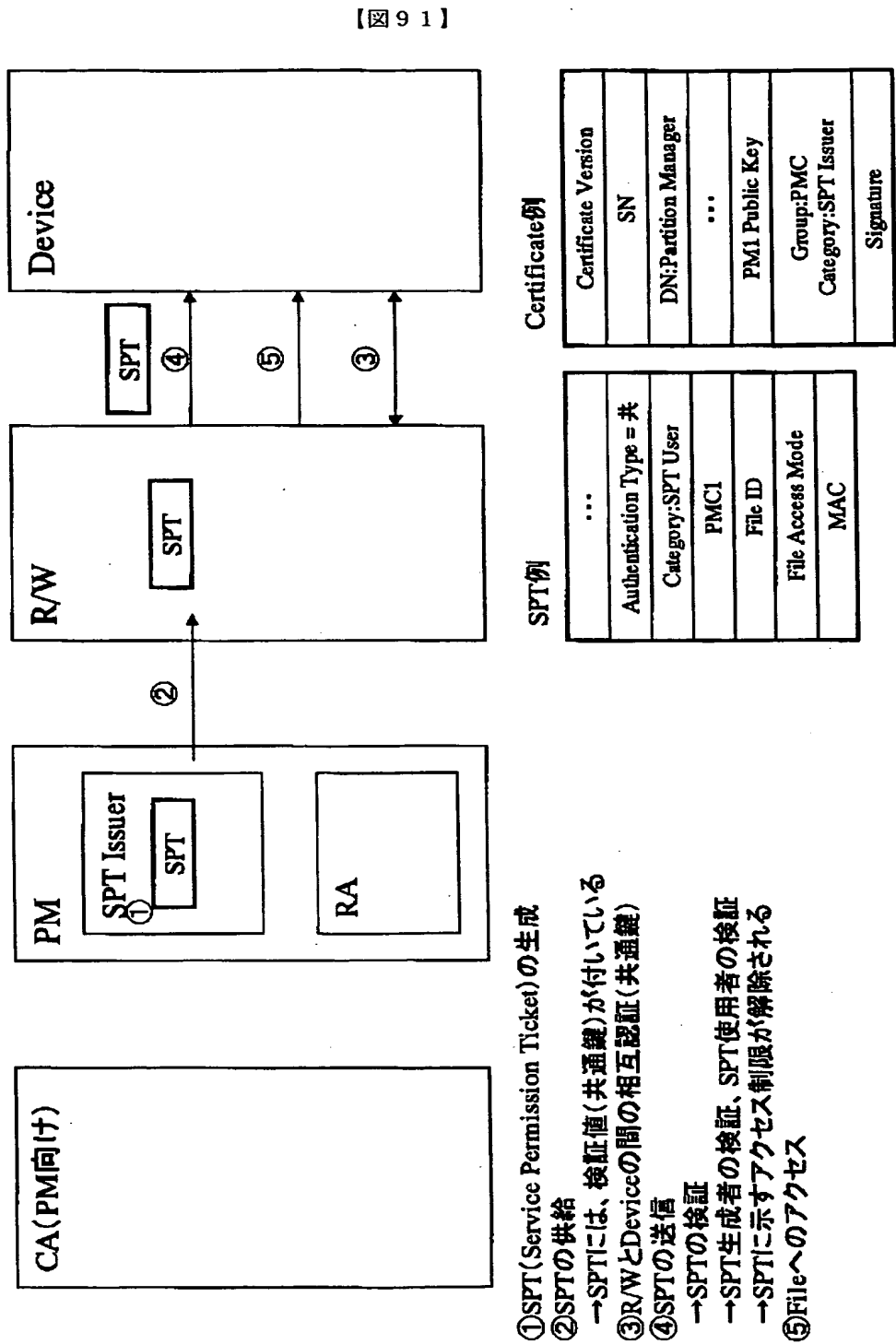
SPT例

...
Authentication Type = 公
Category:SPT User
PMCI
File ID
File Access Mode
MAC

Certificate例

Certificate Version
SN
DN:Partition Manager
...
PM1 Public Key
Group:PMC
Category:SPT Issuer
Signature

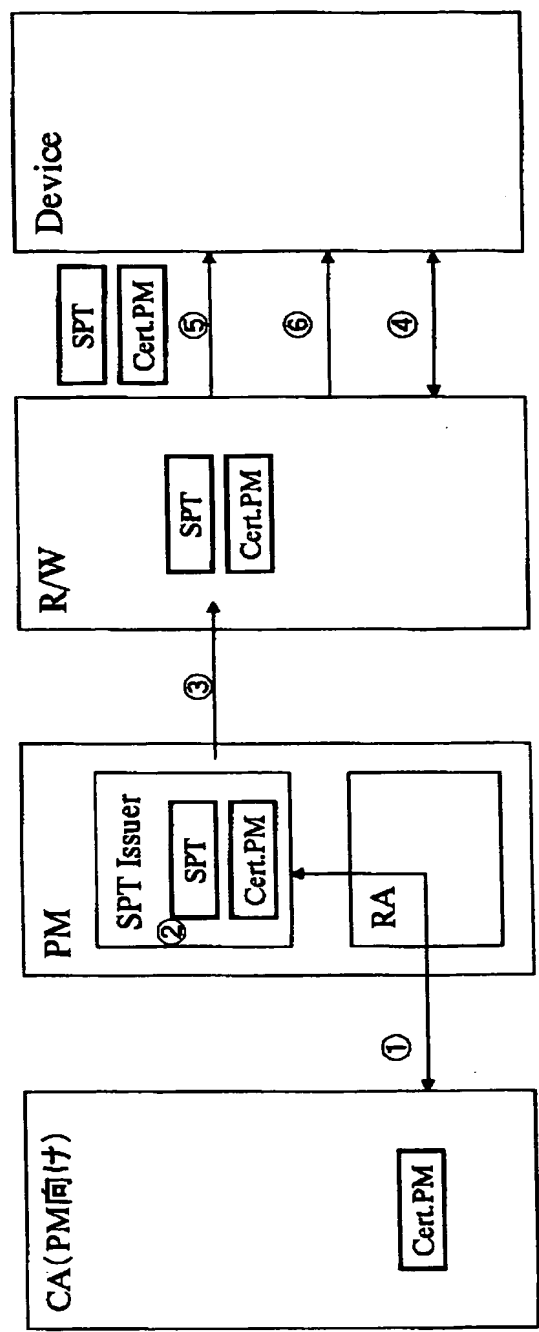
Fileアクセス手順(Authentication Type: 共通鍵, SPT:共通鍵)



【図91】

- ① SPT (Service Permission Ticket) の生成
- ② SPT の供給
  - SPT には、検証値 (共通鍵) が付いている
- ③ R/W と Device の間の相互認証 (共通鍵)
- ④ SPT の送信
  - SPT の検証
  - SPT 生成者の検証、SPT 使用者の検証
  - SPT に示すアクセス制限が解除される
- ⑤ File へのアクセス

Fileアクセス手順(Authentication Type: 共通鍵, SPT:公開鍵)



【図 9 2】

- ①PM用の公開鍵証明書の発行
- ②SPT(Service Permission Ticket)の生成
- ③SPT及びPMのCertificateの供給  
→SPTには、検証値(公開鍵)が付いている
- ④R/WとDeviceの間の相互認証(共通鍵)  
→SPT及びPMのCertificateの送信  
→SPTの検証  
→SPT生成者の検証, SPT使用者の検証  
→SPTに示すアクセス制限が解除される
- ⑥Fileへのアクセス

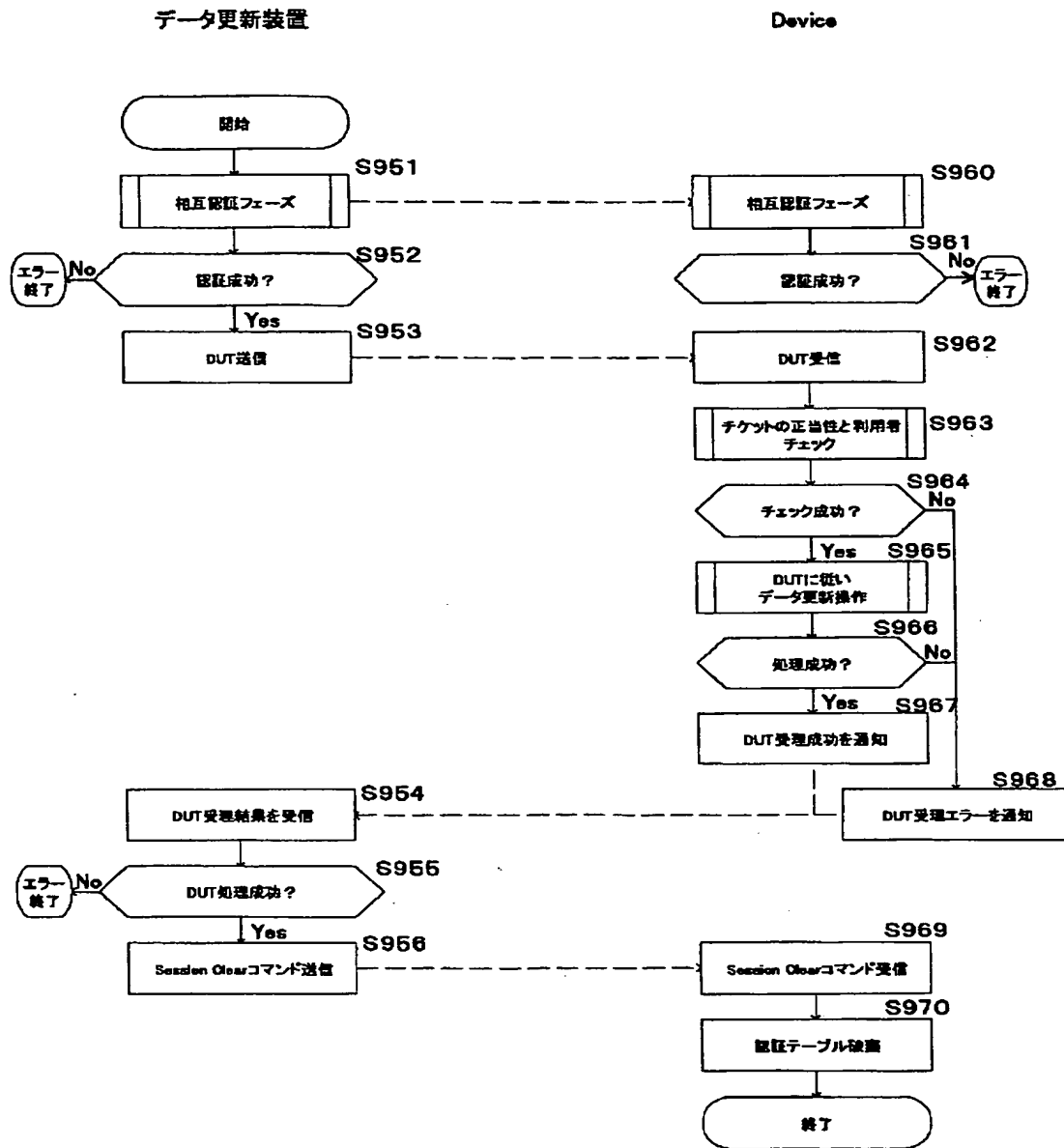
SPT例

...
Authentication Type = 共
Category:SPT User
PMCI
File ID
File Access Mode
Signature

Certificate例

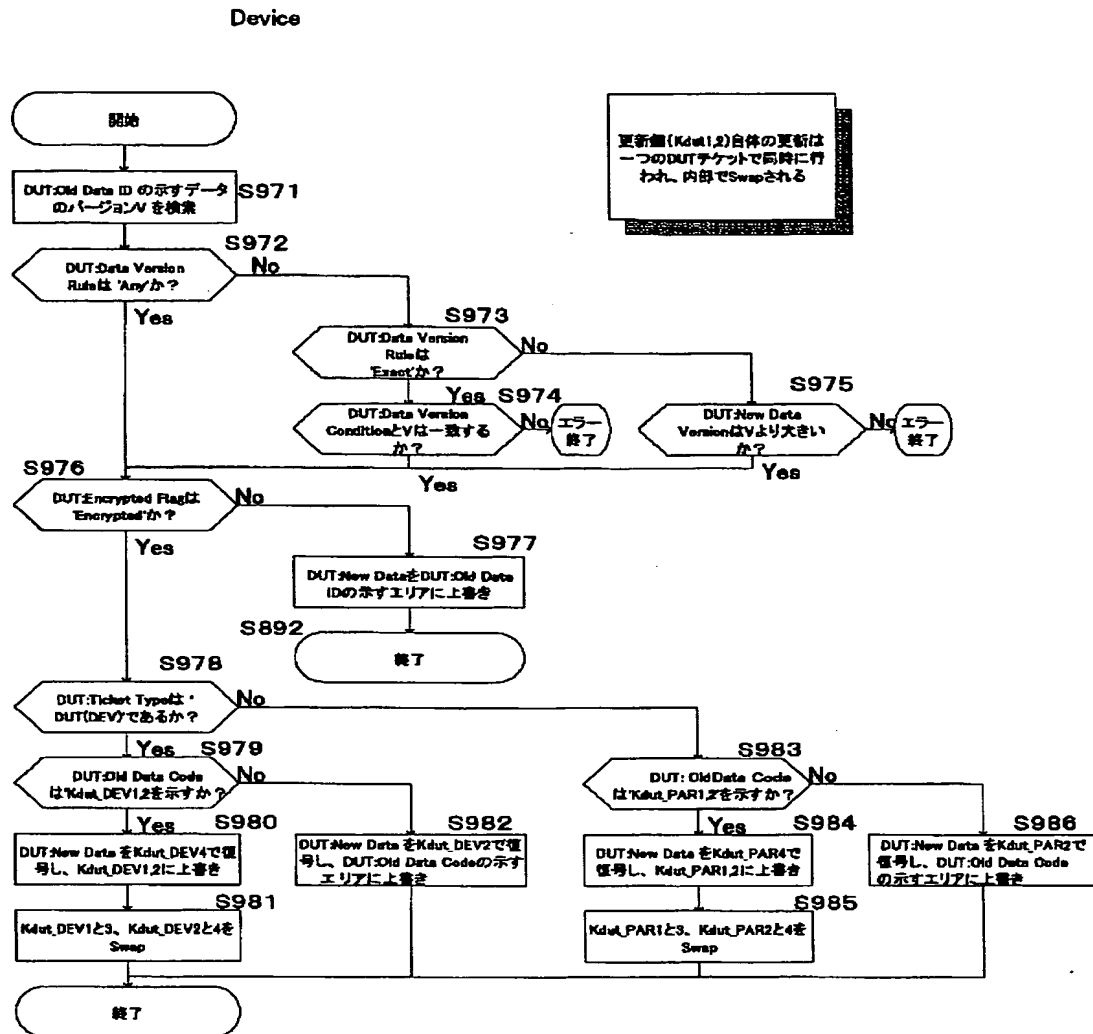
Certificate Version
SN
DN:Partition Manager
...
PM1 Public Key
Group:PMC
Category:SPT Issuer
Signature

【図93】



データ更新

【図94】

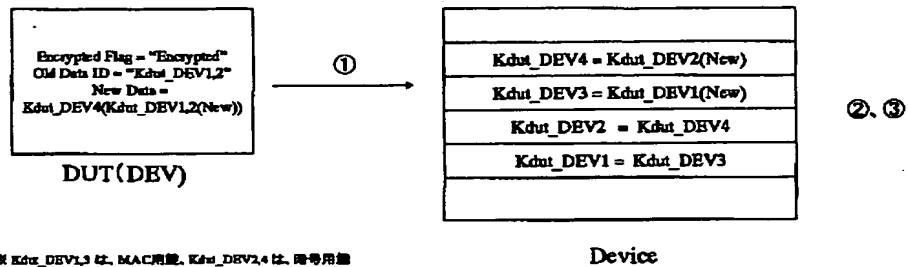


【図95】

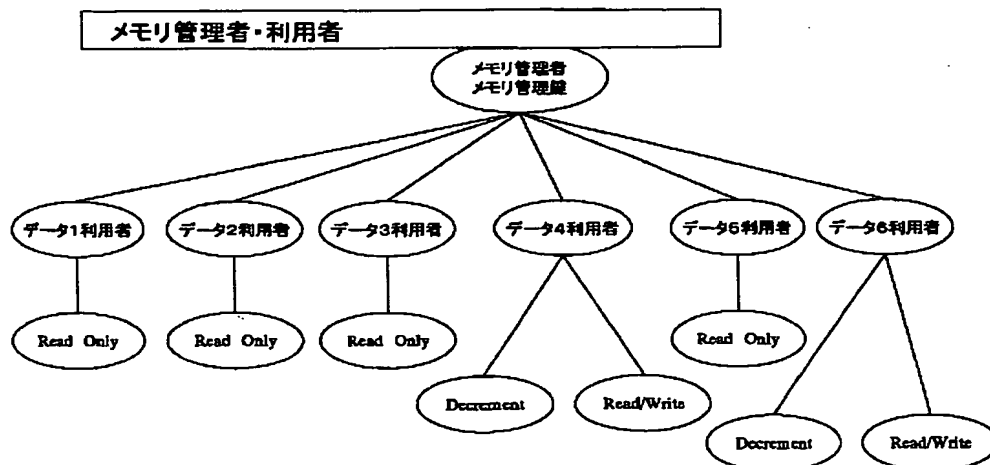
## Data Update

●更新対象が、Kdnt\_DEV1,2 や Kdnt\_PAR1,2 の場合 (Kdnt\_DEV1,2 として説明)

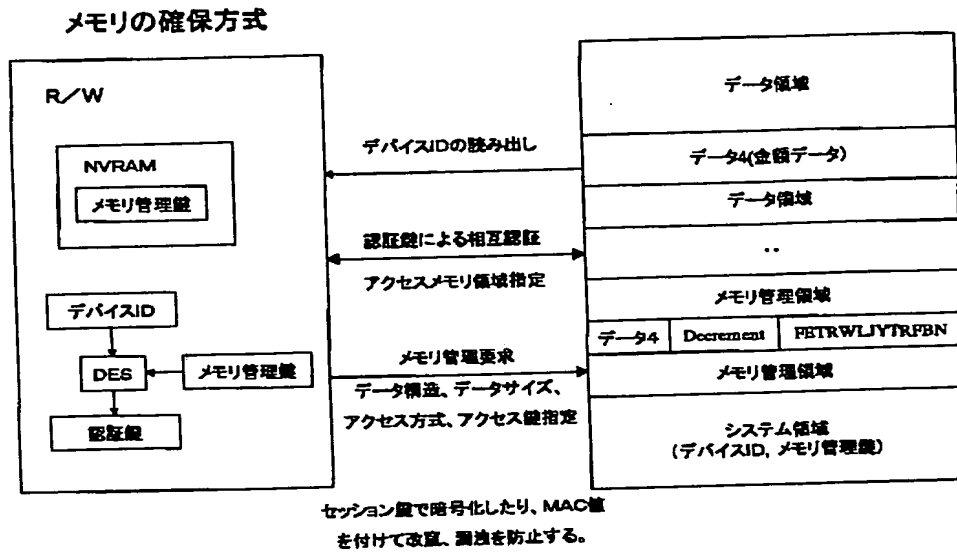
- ① 新しい Kdnt\_DEV1,2 (Kdnt\_DEV1(New), Kdnt\_DEV2(New)) を Kdnt\_DEV4 で暗号化。  
Kdnt\_DEV4(Kdnt\_DEV1(New)), Kdnt\_DEV4(Kdnt\_DEV2(New))  
DUTに記述、Deviceへ送信。
- ② Device は、Kdnt\_DEV1(New), Kdnt\_DEV2(New) を取り出し、古い Kdnt\_DEV1, Kdnt\_DEV2 を上書き
- ③ Kdnt\_DEV1, 3 と Kdnt\_DEV2,4 を入れ替え



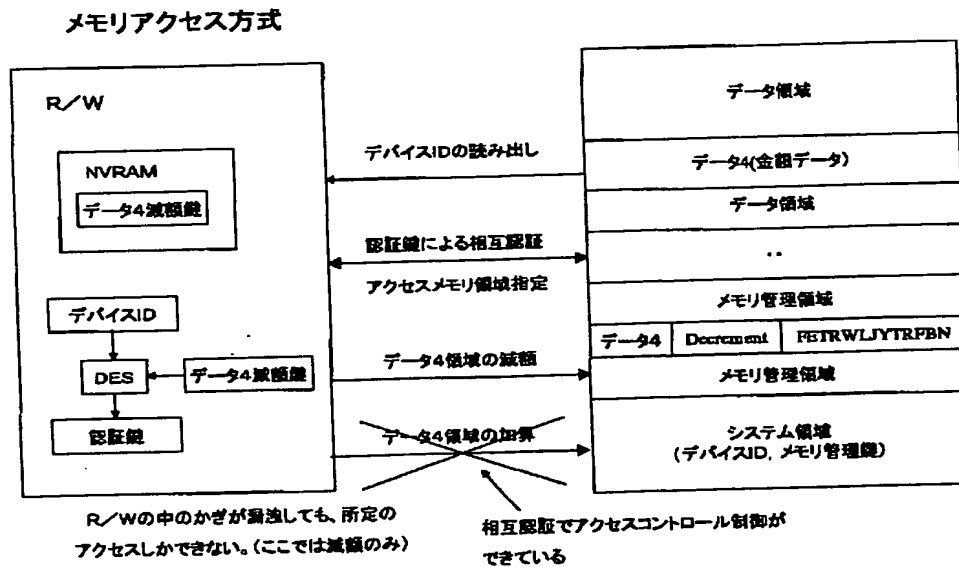
【図97】



【図98】



【図99】



フロントページの続き

(51) Int. Cl.<sup>7</sup>

G 0 6 K 19/10

G 0 9 C 1/00

H 0 4 L 9/32

識別記号

6 4 0

6 6 0

F I

G 0 9 C 1/00

G 0 6 K 19/00

H 0 4 L 9/00

ターマコード(参考)

6 4 0 Z 5 J 1 0 4

6 6 0 C

R

6 7 5 D



(72)発明者 白井 太三  
東京都品川区北品川6丁目7番35号 ソニ  
ー株式会社内  
(72)発明者 高田 昌幸  
東京都品川区北品川6丁目7番35号 ソニ  
ー株式会社内

Fターム(参考) 5B017 AA07 BA06 BA07 BB09 CA11  
CA14  
5B035 AA13 BB09 CA11 CA38  
5B058 CA27 KA02 KA04 KA08 KA31  
KA35 YA20  
5B082 EA11 GA11  
5B085 AA01 AA08 AE01  
5J104 AA07 AA16 EA05 KA02 MA03  
NA02 NA05 NA35 NA38 NA42  
PA12